ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

«Εντοπισμός εξεχουσών τιμών με χρήση τοπικά ευαίσθητου κατακερματισμού(LSH) σε ιεραρχικό επίπεδο»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διονύσιος Καλογηρόπουλος

Εγκρίθηκε απ' την τριμελή επιτροπή:

Δεληγιαννάκης Αντώνιος Επίκουρος Καθηγητής **Επιβλέπων**	Γαροφαλάκης Μίνως Καθηγητής	Σαμολαδάς Βασίλειος Επίκουρος Καθηγητής

8 Νοεμβρίου 2011

ΠΕΡΙΛΗΨΗ

Τα ασύρματα δίκτυα αισθητήρων, που τη σημερινή εποχή χρησιμοποιούνται ευρέως, είναι μία περιοχή συνεχώς αναπτυσσόμενη. Επείδη αντιμετωπίζουν θέματα χαμηλής κατανάλωσης ενέργειας και ασύρματης δικτύωσης μεταξύ τους είναι σημαντική η εύρυθμη λειτουργία τους και η αξιοπιστία των αποτελεσμάτων τους. Στόχος της εργασίας αυτής είναι η εύρεση των ακραίων μετρήσεων του δικτύου χρησιμοποιώντας το σχήμα LSH σε κάθε αισθητήρα. Η εύρεση ακραίων μετρήσεων σ' ένα ασύρματο δίκτυο αισθητήρων είναι σημαντική και έχει ως στόχο την παρατήρηση κάποιου ενδιαφέροντος γεγονότος στο δίκτυο ή την εύρεση κάποιου χαλασμένου αισθητήρα διατηρώντας χαμηλή την κατανάλωση ενέργειας του δικτύου του και επιτυγχάνοντας την εύρυθμη λειτουργία του.

Το ασύρματο δίκτυο αισθητήρων στη τεχνική μας φτιάχνει ένα τοπολογικό δέντρο εξαρτώμενο από την μεταξύ τους απόσταση. Το σχήμα LSH που χρησιμοποιείται βασίζεται στη πιθανολογική μείωση των δεδομένων και αυτό χρησιμεύει σε δεδομένα με μεγάλη διαστατικότητα όπως είναι και το πρόβλημα μας. Το σχήμα αυτό βοηθάει στη γρήγορη αναζήτηση των κοντινών γειτονικών μετρήσεων στο ιστορικό ενός κόμβου και μέσω ενός κατωφλίου που ορίζεται βρίσκονται οι μετρήσεις που θεωρούνται ακραίες για τον αισθητήρα, στέλνοντας στη συνέχεια αυτή τη μέτρηση στο υπόλοιπο δίκτυο. Επίσης στο δίκτυο στέλνεται και ένα μικρό ποσοστό μετρήσεων από κάθε αισθητήρα για να κρατούνται ενήμεροι οι υπόλοιποι αισθητήρες για τα γεγονότα που μπορούν να συμβαίνουν σε άλλα μέρη του δικτύου και ειδικότερα κοντά(τοπολογικά).

Η τεχνική μας με το LSH συγκρίνεται με άλλες δύο τεχνικές που υλοποιήσαμε την άπληστη και τη κεντρικοποιημένη. Οι ποσότητες σύγκρισης είναι η αξιοπιστία, η κατανάλωση ενέργειας και ο χρόνος.

ABSTRACT

Wireless sensor networks are nowadays widely used and form a rapidly growing area of research. Faced with issues of low power consumption and wireless networking, proper operation and reliability of their results are very important. The goal of this work is to find the outliers of a network using the LSH scheme in each sensor. The identification of the extreme measurements in a wireless sensor network is important and its target is to observe some interesting events in the network or to find a damaged sensor, while maintaining low power consumption and proper operation of the network.

In our work, the wireless sensor network creates a topological tree, which is dependent on the distance between the sensors. The LSH scheme used is based on probabilistic data reduction, which is useful for data with high dimensionality as in our problem. This scheme helps to quickly search the neighboring measurements on the history of a node. Through a defined threshold, the measurements which are considered extreme for the sensor can be found and transmitted to the rest of the network. Also a small percentage of measurements from each sensor is sent to the network, so that the rest of the sensors can be kept informed about events that may occur in other parts of the network, especially nearby (topologically).

Our LSH scheme is compared with two other schemes that we implemented, the greedy and the centralized scheme. Reliability, energy consumption and time are our quantities of comparison.

ΑΦΙΕΡΩΣΗ

Θα ήθελα να αφιερώσω την παρούσα εργασία στην οικογένειά μου για την αμέριστη συμπαράσταση, βοήθεια και προ πάντων κατανόηση και ανοχή καθ’ όλο το χρονικό διάστημα των σπουδών μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες σε όλους τους ανθρώπους που συνέβαλαν στο να φέρω εις πέρας την παρούσα Προπτυχιακή Διπλωματική Εργασία. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της εργασίας αυτής κ. Αντώνιο Δεληγιαννάκη για την εμπιστοσύνη που μου έδειξε αναθέτοντας μου αυτή τη διπλωματική εργασία και για την πολύτιμη βοήθεια του και τη διαρκή υποστήριξή του τόσο κατά την ανάπτυξη όσο και κατά τη συγγραφή της παρούσας εργασίας. Επίσης θερμά ευχαριστώ τα μέλη της εξεταστικής επιτροπής μου κ. Μίνωα Γαροφαλάκη και Βασίλειο Σαμολαδά για τις χρήσιμες συμβουλές τους.

Πίνακας περιεχομένων

Κατάλογος Σχημάτων

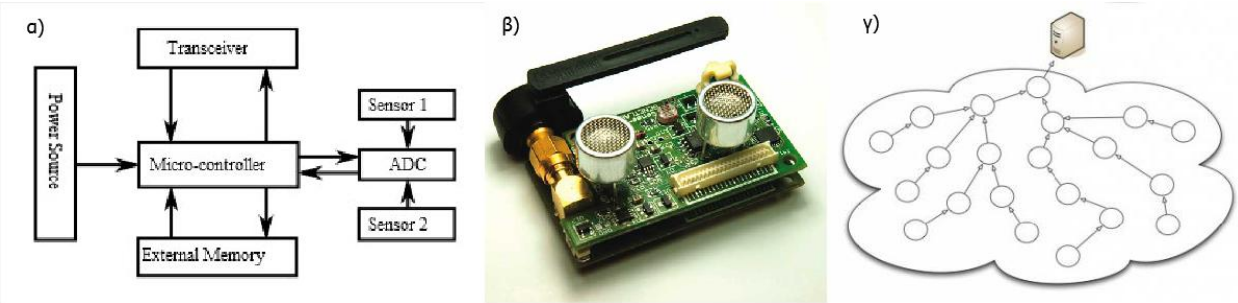
Κατάλογος Εικόνων

Κατάλογος Πινάκων

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Τα δίκτυα αισθητήρων(ΔΑ) με τα οποία ασχολήθηκε η εργασία μας υπάρχουν εδώ και αρκετές δεκαετίες και χρησιμοποιούνται κυρίως για την παρακολούθηση περιβαλλοντικών φαινομένων. Στις μέρες μας τα ΔΑ έχουν προοδεύσει αρκετά και αυτό οφείλεται στην ανάπτυξη της τεχνολογίας έχοντας και έχει ως αποτέλεσμα την παραγωγή μικρότερων αισθητήρων, χαμηλότερο κόστος αγοράς και χαμηλή κατανάλωση ισχύς. Μία από τις πρώτες εφαρμογές που είχαν τα ΔΑ είναι η παρακολούθηση του περιβάλλοντος και η επισήμανση γεγονότων που παρατηρούνται στο περιβάλλον. Οι δυσκολίες όμως που υπάρχουν στην παρακολούθηση σε τέτοια ΔΑ είναι η ύπαρξη περιορισμένων πόρων, η εξέταση δεδομένων του ΔΑ με δυναμικό τρόπο, και η μη-επιβλεπόμενη διαχείρισή τους.

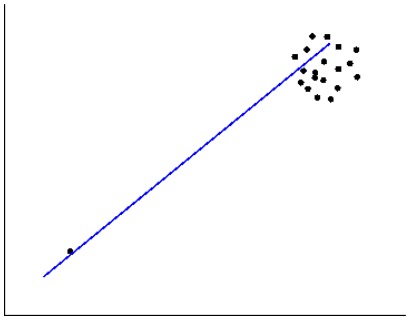


Εικόνα 1.1 α) Βασικά συστατικά ενός αισθητήρα-κόμβου β) ένας ασύρματος αισθητήρα και γ) ένα ΔΑ με ιεραρχική δομή

1.1 Σκοπός της εργασίας

Ο στόχος της εργασίας μας ήταν η διερεύνηση νέων τεχνικών για την ανίχνευση ακραίων τιμών σε ΔΑ. Στον τομέα αυτό, υπάρχει αρκετό ενδιαφέρον από την επιστημονική κοινότητα τα τελευταία χρόνια και όχι μόνο στο χώρο των δικτύων αισθητήρων. Παραδείγματα τέτοιων εργασιών πάνω σε ΔΑ είναι [2][3][4][5]. Υπάρχουν διάφοροι ορισμοί για το τι καθιστά μία τιμή ως ακραία, όπως ο ακόλουθος ορισμός.

Ορισμός 1.1 *Ακραία τιμή(Outlier) ορίζεται η τιμή εκείνη που έχει αφύσικη απόσταση από τις άλλες τιμές του συνόλου δεδομένων.*



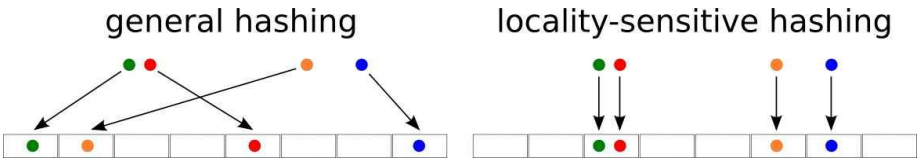
Σχήμα 1.1 Σχηματικό παράδειγμα ακραίας μέτρησης

Συγκεκριμένα ο στόχος της εργασίας μας είναι η ανίχνευση ακραίων μετρήσεων ενός αισθητήρα οι οποίες δεν σχετίζονται αρκετά με άλλες μετρήσεις του ίδιου κόμβου ή και των υπολοίπων του ΔΑ (η απόσταση που πρέπει να έχει η ακραία μέτρηση από τις άλλες ορίζεται στην αρχή ανάλογα με την ευαισθησία που θέλουμε να έχει στη παρατήρηση των γεγονότων του ΔΑ). Μέσω της ανίχνευσης ακραίων τιμών μπορούμε να έχουμε την ανίχνευση κάποιου γεγονότος στο ΔΑ, το φιλτράρισμα πλαστών μετρήσεων ή και την εύρεση ελαττωματικών αισθητηρίων. Άρα βλέπουμε την σημαντικότητα της ανίχνευσης ακραίων μετρήσεων σ'ένα ΔΑ από τις οποίες μπορούμε να εξάγουμε σημαντικές πληροφορίες. Η ανίχνευση θα πρέπει να γίνεται στο δίκτυο αυτοδιαχειριζόμενα για να μην χρειάζεται συνεχής παρακολούθηση από κάποιο άνθρωπο.

Στην εργασία αυτή κοιτάξαμε πως μπορούμε να αντιμετωπίσουμε τις παραπάνω παραμέτρους ώστε να λειτουργεί αποδοτικά το ΔΑ. Μέσω άλλων εργασιών [6][3] βλέπουμε ότι προσεγγιστικές τεχνικές μπορούν να μειώσουν αισθητά το χρόνο απάντησης ενός ερωτήματος, με συνέπεια τη μείωση της επεξεργασίας και σε κάποιες περιπτώσεις τη μείωση της μεταδιδόμενης πληροφορίας άρα και τη μικρότερη κατανάλωση ενέργειας του ΔΑ.

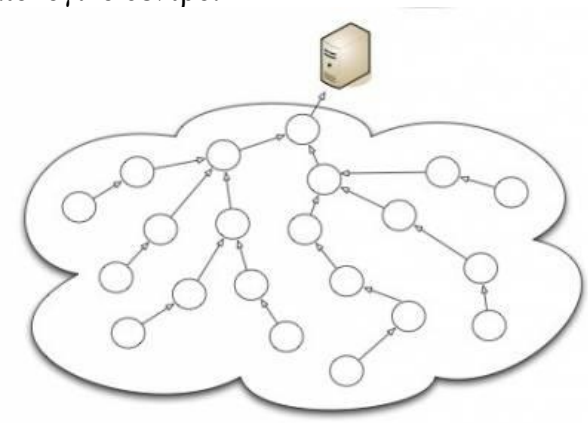
Έτσι προτείνουμε τη χρησιμοποίηση του ευέλικτου σχήματος Locality-Sensitive Hashing(LSH) για την ανίχνευση ακραίων τιμών με την βοήθεια των ευσταθών κατανομών οι οποίες ουσιαστικά πετυχαίνουν τη σύνοψη δεδομένων. Η τεχνική LSH χρησιμοποιείται ευρεία στο πρόβλημα της εύρεσης κοντινότερου γείτονα όπου αναλύεται στη δημοσίευση [1] βρίσκοντας εφαρμογή σε διάφορους τομείς στην επιστήμη των υπολογιστών. Τέτοια παραδείγματα εφαρμογής είναι για αναζήτηση ομοιότητας εικόνας, αναζήτηση ομοιότητας ήχου, ανίχνευση κοντινών διπλότυπων κλπ. Στα ΔΑ η εφαρμογή που μπορεί να έχει όπως είναι και στη δική μας εργασία είναι η εύρεση κοντινών γειτόνων ενός διανύσματος από μετρήσεις ώστε ο αριθμός των κοντινών γειτόνων εάν είναι μεγαλύτερος από ένα κατώφλι να μην χαρακτηρίζεται ακραίο διάνυσμα. Τέτοιες εργασίες που βασίζονται πάνω σε αυτή τη λογική είναι [7] [4]. Το LSH αυξάνει τον απαιτούμενο χώρο που χρειάζεται ο κάθε κόμβος να έχει αλλά μειώνει και το χρόνο απάντησης ενός

ερωτήματος. Επίσης μέσω τη χρησιμοποίηση ενός μικρού δείγματος στην αρχή πετυχαίνουμε τη βέλτιστη απόφαση παραμέτρων της τεχνικής μας (θα εξηγηθεί αναλυτικότερα στο κεφάλαιο 5).



Σχήμα 1.2 Σχηματικό παράδειγμα της τεχνικής LSH σε σχέση με μία απλή τεχνική κατακερματισμού. Τιμές που βρίσκονται κοντά στο χώρο κατακερματίζονται στο ίδιο (ή σε γειτονικούς) κουβά με μεγάλη πιθανότητα.

Αποφασίσαμε να εφαρμόσουμε ιεραρχική οργάνωση στο ΔΑ ώστε να καταφέρουμε την αποκεντροποίηση στην επεξεργασία και απάντηση ερωτημάτων καταφέρνοντας με αυτό το τρόπο το διαμοιρασμό κατανάλωσης ενέργειας σε όλους τους κόμβους του ΔΑ και όχι σε λίγους κόμβους με αποτέλεσμα τη γρήγορη κατανάλωση των πόρων. Έτσι υλοποιήθηκε δομή δέντρου η οποία είναι απλή δομή ως προς την υλοποίηση και θέτει ως ρίζα το σταθμό βάσης. Οι υπόλοιποι κόμβοι τοποθετούνται στο δέντρο ανάλογα με την τοπολογική τους θέση μεταξύ τους. Επομένως το επόμενο επίπεδο μετά την ρίζα θα έχει τους κόμβους οι οποίοι βρίσκονται κάτω από ένα κατώφλι απόστασης από το κόμβο-ρίζα. Αντίστοιχα θα γίνεται αναδρομικά και με τους υπόλοιπους κόμβους δημιουργώντας ένα τοπολογικό δέντρο.



Σχήμα 1.3 Παράδειγμα ενός δέντρου ΔΑ.

Ένας κόμβος στη τεχνική μας χρειάζεται να κρατάει τις απαραίτητες πληροφορίες που χρειάζεται για το δέντρο, ένα δισδιάστατο πίνακα με τα διανύσματα μετρήσεων και ένα πίνακα κατακερματισμού το οποίο κρατάει πληροφορία για τη λειτουργία του LSH σχήματός μας.

Στα παρακάτω κεφάλαια θα εξηγηθούν αναλυτικά οι τεχνικές, οι τρόποι σύγκρισης και τα αποτελέσματα της τεχνικής μας.

1.2 Σύνοψη της εργασίας

Η δομή των κεφαλαίων που ακολουθούν είναι η εξής:

Κεφάλαιο 2: Στο κεφάλαιο αυτό παρουσιάζονται σχετικές εργασίες πάνω στα ΔΑ, σε ανίχνευση ακραίων τιμών και στη τεχνική LSH.

Κεφάλαιο 3: Στο κεφάλαιο αυτό δίνεται η εισαγωγή και η αναλυτική εξήγηση των ευσταθών κατανομών που χρησιμοποιούνται.

Κεφάλαιο 4: Στο κεφάλαιο αυτό δίνεται η εισαγωγή και η αναλυτική εξήγηση της τεχνικής LSH.

Κεφάλαιο 5: Στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά ο σχεδιασμός και η υλοποίηση που γίνεται στην εργασία.

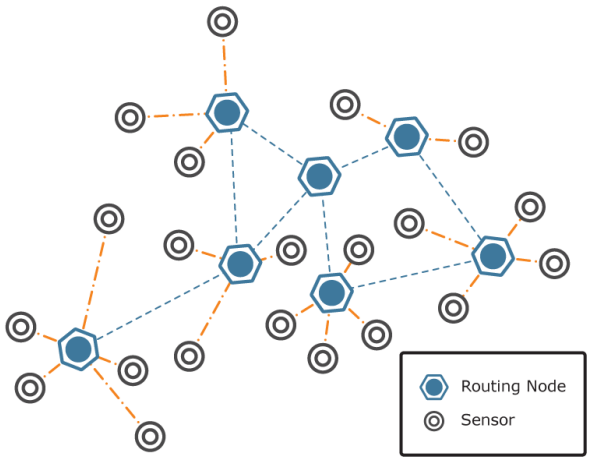
Κεφάλαιο 6: Στο κεφάλαιο αυτό παρουσιάζεται η πειραματική αξιολόγηση της εργασίας μας

ΚΕΦΑΛΑΙΟ 2

ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

2.1 Ασύρματα δίκτυα αισθητήρων

Στην ενότητα αυτή αναφέρονται διάφορες εφαρμογές που μπορούν να έχουν τα ασύρματα δίκτυα αισθητήρων(ΑΔΑ) σε πραγματικά περιβάλλοντα και διάφορες τεχνικές επεξεργασίας και συλλογής δεδομένων. Ένα ΑΔΑ αποτελείται από αυτόνομους αισθητήρες οι οποίοι συνεργάζονται για την παρακολούθηση φυσικών ελέγχων ή περιβαλλοντικών συνθηκών όπως η θερμοκρασία , ήχος , δόνηση , πίεση , κίνηση , ρύπους κλπ. Ο κάθε κόμβος(αισθητήρας) ενός ΑΔΑ αποτελείται από έναν πομποδέκτη , έναν μικροελεγκτή και μια πηγή ενέργειας (*π.χ. η μπαταρία η οποία μπορεί να επαναφορτίζεται με την βοήθεια της ηλιακής ενέργειας*). Επίσης ένα ΑΔΑ αποτελεί συνήθως ένα ασύρματο επί τούτω δίκτυο(*ad-hoc network*) που σημαίνει ότι κάθε αισθητήρας υποστηρίζει δρομολόγηση πολλαπλών αλμάτων(*multi-hop routing*) όπου οι κόμβοι λειτουργούν σαν μεταφορείς και η μεταφορά της πληροφορίας γίνεται σ'ένα σταθμό βάσης. Τέλος σε μια τυπική εφαρμογή οι αισθητήρες είναι διάσπαρτοι σε μια περιοχή και έχουν στόχο να συλλέξουν στοιχεία για αυτό που παρατηρούν.



Σχήμα 2.1 Ασύρματο Δίκτυο Αισθητήρων

Η ανάπτυξη των ΑΔΑ ξεκίνησε από εφαρμογές για στρατιωτικούς σκοπούς και επεκτάθηκε σε ποικίλους τομείς της σημερινής κοινωνίας όπως είναι :

- Αναγνώριση στόχου, κατηγοριοποίηση, παρακολούθηση πορείας
- Παρατήρηση οικοσυστήματος
- Μετεωρολογία
- Παρατήρηση δοκιμής ασφάλειας κτηρίων
- Έξυπνη άρδευση και εκτροφή ζώων
- Παρακολούθηση απορριμμάτων φάρμας
- Παιχνίδια αποφυγής κυνηγού κινδύνου
- Παρακολούθηση υγείας των ανθρώπων
- Έλεγχος λυμάτων και νερού

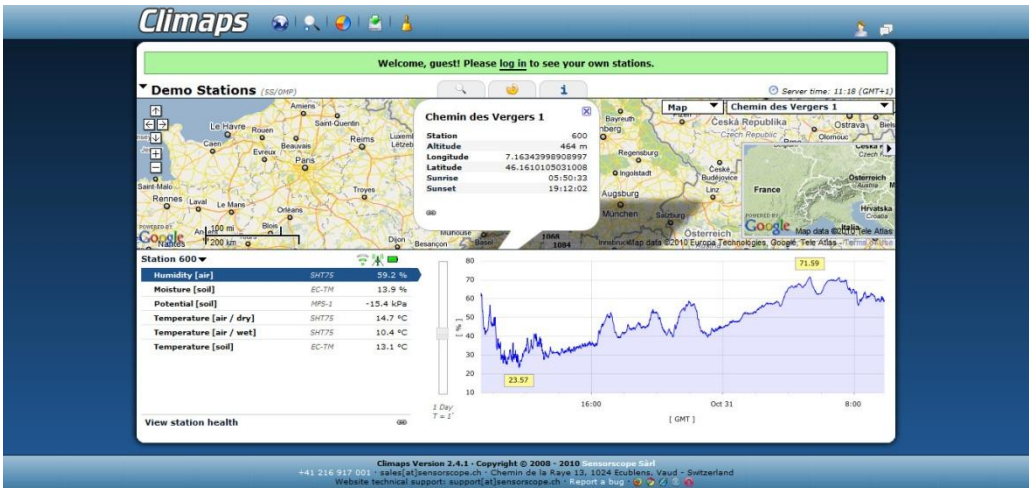
Στις παρακάτω υποενότητες αναφέρονται κάποιες εργασίες σε σχέση με τα ΑΔΑ.

2.1.1 SensorScope

Η εφαρμογή SensorScope[8] χρησιμοποιείται στην παρακολούθηση ενός οικοσυστήματος. Η παρακολούθηση του οικοσυστήματός μας βοηθά στην εξαγωγή χρήσιμων συμπερασμάτων σχετικά με τις κλιματικές αλλαγές , με την επίδρασή τους σ'ένα οικοσύστημα, τη δημιουργία περιβαλλοντικών μοντέλων και την προειδοποίηση επικείμενων κινδύνων όπως *π.χ. χιονοστιβάδες , πλημμύρες κλπ.* Τα ΔΑ χρειάζονται για να μπορούμε να έχουμε μιας μεγάλης κλίμακας κατανεμημένα συστήματα περιβαλλοντικών μετρήσεων που μπορούν να παράγουν μετρήσεις υψηλής χρονικής και χωρικής πυκνότητας. Το σύστημα SensorScope αποτελείται από πολλούς αισθητήρες σταθμοί οι οποίοι μπορούν να μετρήσουν διάφορα περιβαλλοντικά στοιχεία όπως είναι η θερμοκρασία και η υγρασία του αέρα , η θερμοκρασία της επιφάνειας , η εισερχόμενη ηλιακή ακτινοβολία , η ταχύτητα και η διεύθυνση του ανέμου τα επίπεδα βροχόπτωσης και την περιεκτικότητα του εδάφους σε νερό.

Εικόνα 2.1 Αισθητήρες του SensorScope.

Ο σχεδιασμός του συστήματος έγινε με βάση την χαμηλή κατανάλωση ενέργειας, τη μεγάλη εμβέλεια επικοινωνίας, το χαμηλό κόστος και την αντοχή στις καιρικές συνθήκες. Οι σταθμοί λαμβάνουν περιοδικά μετρήσεις και τις διαβιβάζουν στο σταθμό βάσης. Ο σταθμός βάσης προωθεί τις μετρήσεις σ'ένα κεντρικό διακομιστή και μέσω μίας εφαρμογής διαδικτύου ο χρήστης μπορεί να παρακολουθεί τα δεδομένα σε πραγματικό χρόνο.



Εικόνα 2.2 Ιστοπαγής διεπαφή(*Web-based interface*) του συστήματος SensorScope

Το σύστημα SensorScope κάνει χρήση ενός ασύρματου δικτύου αισθητήρων για τη συλλογή των περιβαλλοντικών δεδομένων το οποίο μοιάζει πολύ με τα λεγόμενα επί τούτω δίκτυα. Τέτοια δίκτυα αποτελούνται από αυτόνομες συσκευές οι οποίες λειτουργούν με ένα αυτο-οργανώμενο τρόπο και επικοινωνούν μεταξύ τους με τη χρήση ασύρματων διεπαφών. Επειδή οι σταθμοί πρέπει να έχουν χαμηλή κατανάλωση ενέργειας και λόγω των περιορισμών από τις φυσικές ιδιότητες των ραδιοκυμάτων οι σταθμοί μπορούν να επικοινωνούν με σταθμούς οι οποίοι βρίσκονται σε μικρή απόσταση. Αυτό συνεπάγεται ότι το δίκτυο πρέπει να χρησιμοποιεί δρομολόγηση πολλαπλών αλμάτων για να μπορέσουν να φτάσουν οι μετρήσεις από τους κόμβους στο σταθμό βάσης.

Το SensorScope χρησιμοποιεί ένα σταθμό βάσης για τη συλλογή όλων των δεδομένων. Επίσης έχει τη δυνατότητα να τροποποιεί την παρακολουθούμενη περιοχή, εάν υπάρχει μετακίνηση , πρόσθεση ή διαγραφή σταθμού όποτε χρειαστεί. Το SensorScope για να κάνει τις αλλαγές αυτές δεν χρειάζεται να κάνει ανασυγκρότηση του δικτύου του. *Π.χ. εάν ένας σταθμός σταματήσει να λειτουργεί δεν επηρεάζει τη συλλογή των δεδομένων. Ακόμα και στην περίπτωση που ήταν μέρος της διαδρομής για το σταθμό βάσης τότε το δίκτυο δημιουργεί αυτόματα ένα νέο μονοπάτι.*

Για την μείωση της κατανάλωσης ενέργειας ο σταθμός έχει την δυνατότητα να απενεργοποιεί τον ασύρματο πομπό και δέκτη(έχει μεγάλη κατανάλωση ενέργειας) χωρίς να έχει τις περισσότερες φορές επίδραση στη συλλογή των δεδομένων. Εξαιτίας της παραπάνω δυνατότητα και λόγω του γεγονότος ότι οι σταθμοί έχουν σύστημα συλλογής ηλιακής ενέργειας θεωρητικά δεν θα μπορούν να σταματήσουν να λειτουργούν λόγω έλλειψης ενέργειας.

2.1.2 CodeBlue: Ασύρματοι αισθητήρες για ιατρική φροντίδα

Τα ΑΔΑ επίσης μπορούν να εφαρμοστούν σε διάφορες ιατρικές εφαρμογές όπως για προ-νοσοκομειακή φροντίδα , για την αποκατάσταση των ασθενών μετά από εγκεφαλικό επεισόδιο και ποικίλες ακόμα εφαρμογές. Με την βοήθεια των ΑΔΑ η τεχνολογία CodeBlue[9] επιτρέπει την αυτόματη παρακολούθηση ζωτικών σημείων του ασθενούς , (την καταγραφή της εξέλιξης της υγείας του ασθενή σε πραγματικό χρόνο καθώς και την δυνατότητα συσχέτισης με άλλα αρχεία των νοσοκομείων για σύγκριση και ασφαλέστερα συμπεράσματα.

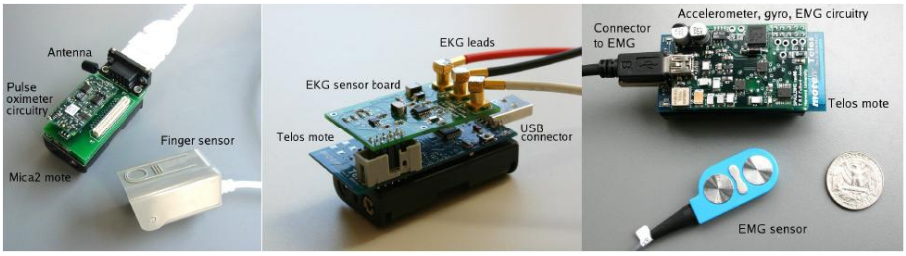
2.1.2.1 Χαρακτηριστικά του συστήματος CodeBlue

Όσον αφορά τα χαρακτηριστικά που πρέπει να έχει ένα ΑΔΑ όπως το CodeBlue που αφορά ιατρικό σχεδιασμό αυτά θα εξαρτώνται και από την ειδική εφαρμογή και το περιβάλλον ανάπτυξης του. *Π.χ. ένα ΑΔΑ που είναι σχεδιασμένο για επί τούτο δίκτυο σε μία κατάσταση έκτατης ανάγκης έχει αρκετά διαφορετικές απαιτήσεις από μία μόνιμη εγκατάσταση σ'ένα νοσοκομείο. Στη μόνιμη εγκατάσταση μπορούμε να έχουμε σταθερούς ηλεκτρικούς κόμβους παρέχοντας σύνδεση σ'ένα ενσύρματο δίκτυο υποδομής.* Τα χαρακτηριστικά του συστήματος CodeBlue είναι:

- Οι αισθητήρες είναι πολύ μικροί , ελαφροί και μπορούν να φορεθούν εύκολα.
- Η επικοινωνία των αισθητήρων με το σταθμό βάσης θα είναι γενικά αξιόπιστη γιατί δεν επιτρέπεται να έχουμε συνεχόμενη απώλεια δεδομένων.
- Υποστήριξη multicast για να μπορούν να λαμβάνουν τις μετρήσεις ενός ασθενούς διαφορετικοί γιατροί παράλληλα.
- Ασφάλεια των δεδομένων στο σύστημα(απόρρητο).
- Υποστήριξη κινητικότητας τόσο των ασθενών όσο και των γιατρών.

Το CodeBlue υποστηρίζει τρεις ασύρματους αισθητήρες 1) παλμικό οξύμετρο 2) ηλεκτροκαρδιογραφητή και 3) έναν αναλυτή κίνησης. Το παλμικό οξύμετρο μετράει την καρδιακή συχνότητα και την περιεκτικότητα του αίματος σε οξυγόνο. Ο ηλεκτροκαρδιογραφητής παρατηρεί την ηλεκτρική δραστηριότητα της καρδιάς και ο

αναλυτής κίνησης παρατηρεί την μυϊκή δραστηριότητα και το σκέλος των κινήσεων του ασθενούς.



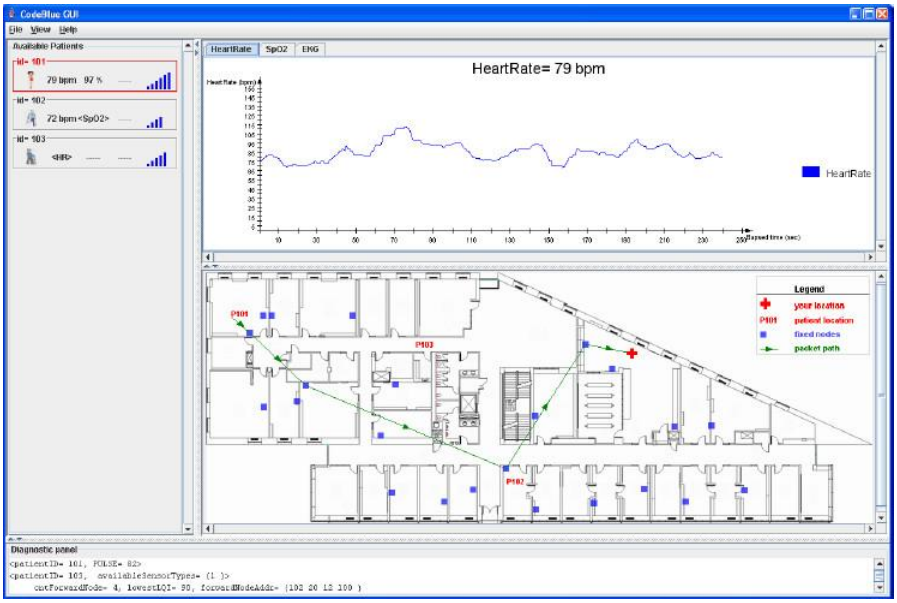
Εικόνα 2.3 Αισθητήρες του συστήματος CodeBlue

2.1.2.2 Περιγραφή σχεδιασμού και αρχιτεκτονικής CodeBlue

Το CodeBlue είναι υλοποιημένο σε «TinyOS»[16] και παρέχει πρωτόκολλα για την ενσωμάτωση των ασύρματων ιατρικών αισθητήρων και συσκευών τελικού χρήστη όπως μπορεί να είναι τα «PDAs» και φορητοί υπολογιστές. Το CodeBlue βασίζεται στη δρομολόγηση πλαισίου που επιτρέπει πολλαπλές συσκευές αισθητήρων να αναμεταδίδουν τα δεδομένα σε όλους τους παραλήπτες που έχουν εκδηλώσει ενδιαφέρον να λαμβάνουν αυτά τα δεδομένα. Αυτό το μοντέλο ταιριάζει σε ιατρικές εφαρμογές που θα χρειάζεται διάφοροι φροντιστές να παρακολουθούν τις μετρήσεις ενός ασθενή.

Οι αισθητήρες δημοσιεύουν σχετικά δεδομένα σ'ένα συγκεκριμένο κανάλι και η συσκευή τελικού χρήστη κάνει εγγραφή σε κανάλια που τους ενδιαφέρουν. Το μοντέλο του CodeBlue πρέπει να επισημανθεί ότι:

- Οι αισθητήρες δεν μπορούν να δημοσιεύουν δεδομένα σε μία αυθαίρετη τιμή γιατί το κανάλι έχει περιορισμένο εύρος ζώνης
- Με δεδομένο ότι οι εκδότες και οι συνδρομητές μπορεί να μην είναι εντός της εμβέλειας τους θα πρέπει να υπάρχει κάποια μορφή δρομολόγησης
- Το επίπεδο της επικοινωνίας θα πρέπει να λάβει υπόψη την κινητικότητα κατά τη θέσπιση δρομολόγησης ενός μονοπατιού(ασθενείς και φροντιστές είναι κινητοί)
- Το επίπεδο δρομολόγησης της CodeBlue βασίζεται στο πρωτόκολλο *Driven Multicast Routing (ADMR)*



Εικόνα 2.4 Η διεπαφή χρήστη στο σύστημα CodeBlue. Είναι ένα στιγμιότυπο από μία πραγματική εφαρμογή του CodeBlue GUI που τρέχει σ'ένα κτίριο με τρεις αισθητήρες-ασθενή και αναφέρονται τα δεδομένα σ'ένα φορητό υπολογιστή.

2.1.3 Ασύρματα δίκτυα αισθητήρων για παρακολούθηση οικοσυστήματος

Οι βιοεπιστήμονες ανησυχούν για τις επιπτώσεις που έχει η ανθρώπινη παρουσία στην παρακολούθηση φυτών και ζώων. Το σύστημα αυτό με την βοήθεια των ασύρματων δικτύων αισθητήρων θέλει να εξαλείψει την ανθρώπινη παρουσία για την παρακολούθηση ενός ευαίσθητου οικοσυστήματος.

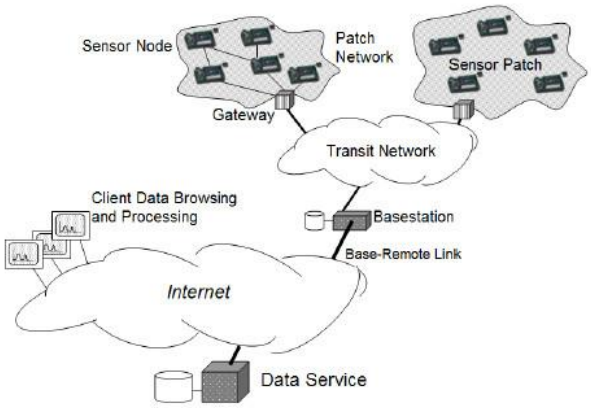
Το πανεπιστήμιο Atlantic έχει τομέα που ασχολείται με τα ΔΑ για την παρακολούθηση οικοσυστημάτων. Ένα από τα απομακρυσμένα οικοσυστήματα όπου έχει εγκατασταθεί το σύστημα που θα αναφέρεται βρίσκεται στο νησί Μεγάλη Πάπια(*Great Duck*) το οποίο ανήκει στο κράτος του Μάιν(*Maine*). Με το παραπάνω ΔΑ παρακολουθείται το πτηνό θαλασσοβάτης για τους παρακάτω λόγους:

- Ο τρόπος χρήσης των φωλιών στο κύκλο από 24-72 ώρες όταν ένα ή και τα δύο μέλη του ζεύγους αναπαραγωγής εναλλάσσουν τα καθήκοντα επώασης με τη διατροφή στην θάλασσα.
- Οι αλλαγές που υπάρχουν στην φωλιά και στις περιβαλλοντικές παραμέτρους κατά την αναπαραγωγική περίοδο (Απρίλιος- Οκτώβριος)
- Οι διαφορές που υπάρχουν στα περιβάλλοντα με ή χωρίς μεγάλο αριθμό από φωλιές του είδους θαλασσοβάτης.

2.1.3.1 Χαρακτηριστικά του συστήματος για το νησί Great Duck

Τα χαρακτηριστικά του συστήματος για την παρακολούθηση του παραπάνω οικοσυστήματος είναι:

- Τα ΔΑ θα πρέπει να είναι προσβάσιμα μέσω του διαδικτύου
- Ιεραρχικό δίκτυο (φαίνεται στο παρακάτω σχήμα που παρατίθεται)
- Μακροβιότητα του δικτύου αισθητήρων (ΔΑ που κινούνται για 9 μήνες χωρίς επαναφορτιζόμενες μπαταρίες έχουν μεγάλες απαιτήσεις)
- Κάθε επίπεδο του δικτύου πρέπει να λειτουργεί με συγκεκριμένη διαθέσιμη ενέργεια
- Διαχείριση των αισθητήρων από απόσταση(μέσω διαδικτύου)
- Η παρακολούθηση του οικοσυστήματος θα πρέπει να είναι δυσδιάκριτη
- Τα ΔΑ του συστήματος να παρουσιάζουν σταθερή και προβλέψιμη συμπεριφορά
- Μέτρηση της έντασης του φωτός , θερμοκρασία , υπέρυθρες , σχετική εργασία και βαρομετρική πίεση
- Αρχειοθέτηση των μετρήσεων των αισθητήρων για εύκολη εξόρυξη και ανάλυση δεδομένων



Σχήμα 2.2 Αρχιτεκτονική συστήματος για παρακολούθηση του οικοσυστήματος

2.1.3.2 Περιγραφή αρχιτεκτονικής συστήματος

Το σύστημα έχει μία κλιμακωτή αρχιτεκτονική. Ξεκινάει από το χαμηλότερο επίπεδο που αποτελείται από ένα μεγάλο αριθμό αυτόνομων φτηνών αισθητήρων (υψηλή χωρική ανάλυση) όπου ο καθένας συλλέγει περιβαλλοντικά δεδομένα για το άμεσο

περιβάλλον του και είναι υπεύθυνο για την επικοινωνία με τους άλλους κόμβους. Η επικοινωνία με τους άλλους κόμβους αισθητήρες γίνεται σ΄ένα δίκτυο που υποστηρίζει πολλαπλά άλματα άρα το σύστημα μπορεί να εξαγάγει και συναθροιστικά αποτελέσματα. Στο επόμενο επίπεδο η πύλη της κάθε ομάδας από αισθητήρες διαβιβάζει τα δεδομένα τους μέσω του δικτύου μεταφοράς στο σταθμό βάσης. Το επόμενο επίπεδο που βρίσκεται ο σταθμός βάσης συνδέεται με αντίγραφα των δεδομένων μέσω του διαδικτύου και επίσης γίνεται το φιλτράρισμα και η επεξεργασία των δεδομένων που έχει λάβει από τους αισθητήρες. Στο υψηλότερο επίπεδο βρίσκονται οι χρήστες-επιστήμονες οι οποίοι μπορούν να διαβάσουν τα δεδομένα μέσω της διεπαφής.

2.2 Ανίχνευση ακραίων τιμών σε δίκτυα αισθητήρων

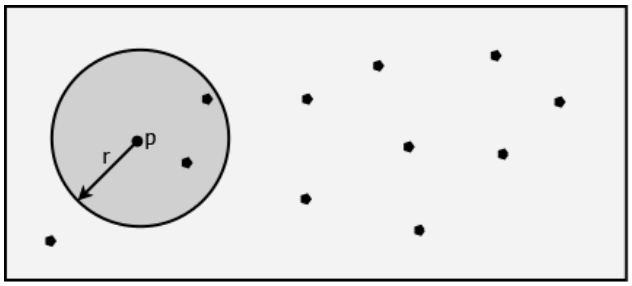
Στην ενότητα αυτή αναφέρουμε σχετικές εργασίες που έχουν γίνει για την ανίχνευση ακραίων τιμών σε ΔΑ. Ο εντοπισμός των ακραίων τιμών είναι σημαντικός γιατί μπορεί να υποδηλώνει ελαττωματικό αισθητήρα ή την ανίχνευση κάποιου ενδιαφέροντος γεγονότος. Λόγω του φθηνού υλικού κατασκευής τους και της ευαισθησίας των αισθητηρίων οργάνων σε περιβαλλοντολογικές συνθήκες (όπως η υγρασία), οι αισθητήρες συχνά είτε λαμβάνουν «περίεργες» μετρήσεις, είτε παρουσιάζουν το φαινόμενη της βρώμικης αποτυχίας(*fail-dirty*). Σε αυτή την περίπτωση, ο αισθητήρας αυξάνει/μειώνει σταδιακά τη μέτρησή του μέχρι αυτή να φτάσει σε μία μέγιστη/ελάχιστη τιμή. Συνεπώς, ο εντοπισμός ακραίων τιμών μπορεί να βοηθήσει στον εντοπισμό αισθητήρων που έχουν βρώμικες μετρήσεις. Από την άλλη, μία ακραία τιμή μπορεί να συνιστά την απαρχή ενός ενδιαφέροντος γεγονότος. Για παράδειγμα, υψηλές τιμές θερμοκρασίας μπορεί να οφείλονται στην ύπαρξη κάποιας φωτιάς. Συνεπώς, για εφαρμογές επιτήρησης είναι πολύ σημαντικό να εντοπίζονται αυτόματα ακραίες τιμές, και να λαμβάνεται περαιτέρω δράση σε αντίστοιχους εντοπισμούς. Σχετικές εργασίες που εντοπίζουν ακραίες τιμές πάνω σε ΔΑ βασίζονται σε διαφορετικούς ορισμούς της ακραίας τιμής και σε διαφορετικές τεχνικές εντοπισμού τους και αντιμετώπισή τους.

2.2.1 Online Outlier Detection in Sensor Data Using Non-Parametric-Models

Η δημοσίευση [3] δίνει 2 εναλλακτικούς ορισμούς για το πότε μία μέτρηση ενός αισθητήρα χαρακτηρίζεται ως ακραία τιμή – ο πρώτος ορισμός είναι με βάση την απόσταση και ο δεύτερος με βάση την πυκνότητα.

Ορισμός 2.1 *Βάση της απόστασης ένα σημείο p σε ένα σύνολο T είναι ακραία τιμή αν το πολύ D από τα σημεία του T βρίσκονται σε απόσταση το πολύ r από το p.*

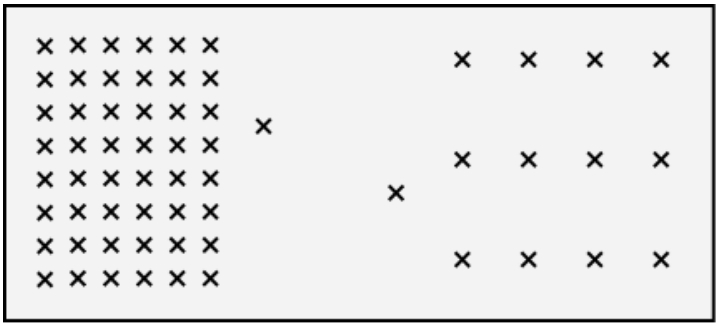
π.χ. εάν το D(κατώφλι για να θεωρείται ακραία τιμή) είναι 5 και το σημείο p έχει 4 σημεία του T σε ακτίνα r τότε το σημείο p θεωρείται ακραία τιμή.



Σχήμα 2.3 Ορισμός ακραίας τιμής με βάση την απόσταση.

Ορισμός 2.1 *Βάση την πυκνότητα ακραίες τιμές είναι οι τιμές για τις οποίες η πυκνότητα της κοντινής γειτονιάς τους είναι σημαντικά μικρότερη της ευρύτερης περιοχής τους*

π.χ. Για μία νέα παρατήρηση p , ένας αισθητήρας μπορεί να χρησιμοποιήσει το μοντέλο εκτίμησης πυκνότητας για να καθορίσει αν η p είναι μία MDEF-ακραία τιμή όσον αφορά την ροή δεδομένων του.



Σχήμα 2.4 Ορισμός ακραίας τιμής με βάση την πυκνότητα.

Η τεχνική που χρησιμοποιείται για την ανίχνευση ακραίων τιμών βασίζεται σε ένα κατανεμημένο σύστημα (ιεραρχική δομή των αισθητήρων - χωρισμός σε επίπεδα) όπου ο κάθε αισθητήρας διατηρεί ένα τυχαίο δείγμα από τις μετρήσεις που έχουν ληφθεί στο υποδέντρο του. Η διαδικασία που ακολουθεί ο αλγόριθμος είναι:

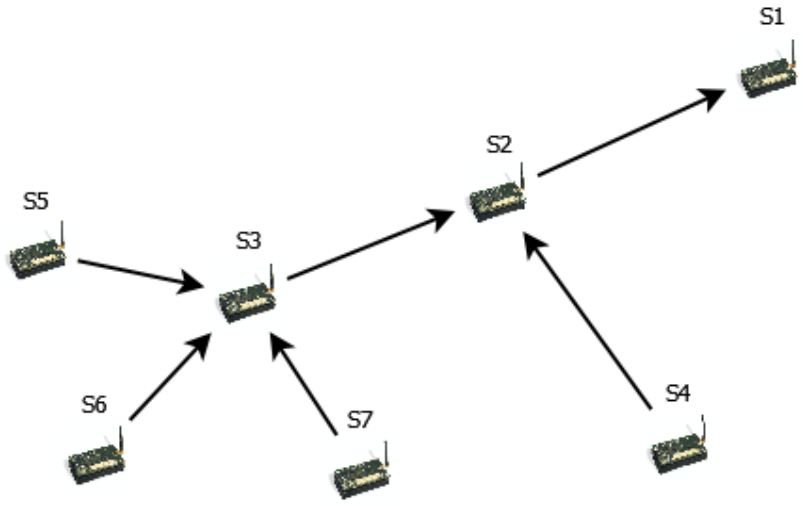
- Οι κόμβοι φύλλα αποφασίζουν εάν υπάρχουν ακραίες τιμές στις μετρήσεις τους. Αν ναι, στέλνουν τη μέτρηση στον πατέρα τους.

- Επιπλέον, η μέτρηση μπορεί να εισαχθεί στο τυχαίο δείγμα που διατηρεί ο κόμβος, με μία καθορισμένη πιθανότητα p προωθούν τη μέτρηση στο παραπάνω επίπεδο (στον «ηγέτη κόμβο»).
- Κάθε εσωτερικός κόμβος του δέντρου, όταν λαμβάνει από ένα παιδί του μία μέτρηση ως πιθανή ακραία τιμή, πραγματοποιεί τον αντίστοιχο έλεγχο με βάση το δικό του τυχαίο δείγμα (το οποίο αντιστοιχεί σε μετρήσεις μεγαλύτερου τμήματος του δικτύου). Αν πάλι κριθεί η μέτρηση ως πιθανή ακραία τιμή, τότε πραγματοποιείται η προώθηση της τιμής στον πατέρα του κόμβου κτλ.
- Η διαδικασία αυτή επαναλαμβάνεται έως ότου φτάσει στο υψηλότερο επίπεδο την ρίζα, που αποφασίζεται εάν μια μέτρηση είναι πραγματικά ακραία τιμή με βάση τον ορισμό.

Η επεξεργασία των δεδομένων γίνεται σε πραγματικό χρόνο και για αυτό το σκοπό χρησιμοποιείται ένα προσεγγιστικό μοντέλο ο πυρήνας εκτιμητής πυκνότητας(*kernel density estimator*). Ο κάθε αισθητήρας με τη χρήση του παραπάνω εκτιμητή κρατάει ένα μοντέλο για την κατανομή των μετρήσεων που παράγει. Ειδικά για την περίπτωση ανίχνευσης ακραίων τιμών με βάση την πυκνότητα, ο κάθε αισθητήρας χρειάζεται μία εκτίμηση της συνολικής πυκνότητας (για όλο το δίκτυο) γύρω από κάθε πιθανή μέτρηση. Για αυτό, και μόνο στην ανίχνευση ακραίων τιμών με βάση την πυκνότητα, το αντίστοιχο μοντέλο της ρίζας του δέντρου μεταδίδεται από τη ρίζα προς όλους τους κόμβους σε κάθε ενημέρωση του.

2.2.2 Another Outlier Bites the Dust: Computing Meaningful Aggregates in Sensor Networks

Μία άλλη δημοσίευση[2] για την ανίχνευση ακραίων τιμών υποστηρίζει συναθροιστικά ερωτήματα όπως «μέγιστο», «ελάχιστο», «αρίθμηση», «σύνολο», «μέσος όρος» και ο σκοπός είναι ο υπολογισμός ενός «καθαρού» συναθροιστικού αποτελέσματος, μαζί με ένα σύνολο χαρακτηριστικών τιμών, όπως και ένα σύνολο ακραίων τιμών. Σε αυτή τη δημοσίευση η μέτρηση ενός αισθητήρα χαρακτηρίζεται ως ακραία τιμή αν το μοτίβο των πρόσφατων μετρήσεων του αισθητήρα μοιάζει το πολύ με αυτό D άλλων κόμβων. Η τεχνική που ακολουθεί η δημοσίευση για τον υπολογισμό των συναθροιστικών συναρτήσεων και για την ανίχνευση ακραίων τιμών βασίζεται σε ένα δέντρο συλλογής το οποίο μπορεί να προσαρμόζεται ανάλογα με τα στατιστικά στοιχεία που συλλέγει για να έχει μεγαλύτερη αξιοπιστία.



Σχήμα 2.5 Δέντρο συλλογής.

Ένας κόμβος χαρακτηρίζεται ως ακραία τιμή εάν έχει λιγότερους υποστηρικτές από το κατώφλι υποστηρικτών που ορίζει ο χρήστης σε κάθε ερώτημα. Υποστηρικτής ενός κόμβου ορίζεται από την μετρική ομοιότητα που έχουν μεταξύ τους οι δύο κόμβοι. Στην δημοσίευση για την μετρική ομοιότητα των διανυσμάτων x_i, x_j τιμών δύο κόμβων S_i και S_j, χρησιμοποιούνται οι παρακάτω τρόποι:

1. Συντελεστής συσχέτισης

2. Συντελεστής Jaccard

3. Γραμμική παλινδρόμηση (πόσο καλά προσεγγίζονται οι μετρήσεις του ενός αισθητήρα με βάση τις μετρήσεις του άλλου).

Για την αποφυγή λανθασμένων αποτελεσμάτων στα συναθροιστικά ερωτήματα, όταν μια τιμή χαρακτηρίζεται ως ακραία τιμή, αυτή δεν καταμετράται στο ερώτημα, άλλα στέλνεται στα παραπάνω επίπεδα του δέντρου, για να εξεταστεί, εάν θεωρείται ακραία τιμή και στο υπόλοιπο δέντρο. Για να μπορούν να γίνουν οι συγκρίσεις με βάση το μοτίβο του κάθε κόμβου θα πρέπει να κρατάει κάθε κόμβος στην μνήμη τις τελευταίες μετρήσεις κάποιων απογόνων του. Το δέντρο συλλογής περιοδικά αναδιοργανώνεται έτσι ώστε οι μετρήσεις κάθε κόμβου να κατευθύνονται προς κοντινούς κόμβους όπου αναμένεται να βρει ο αισθητήρας πιο εύκολα υποστήριξη για τη μέτρησή του.

2.2.3 Using SensorRanks for In-Network Detection of Faulty Readings in Wireless Sensor Networks

Τέλος, άλλη μία δημοσίευση[5] για την ανίχνευση ακραίων τιμών εστιάζει στην ανίχνευση ακραίων τιμών που μπορεί να οφείλονται σε προβληματικούς αισθητήρες, οι οποίοι παράγουν αυθαίρετες αναγνώσεις ή λόγων παρεμβολών μπορεί να παράγουν αναγνώσεις που έχουν θόρυβο. Ο αλγόριθμος για τον εντοπισμό των ακραίων τιμών βασίζεται στην κατασκευή δικτύου συσχετίσεων όπου οι γειτονικοί κόμβοι συνδέονται με ακμές και το βάρος ακμής υποδηλώνει το πόσο όμοιες είναι μεταξύ τους οι μετρήσεις τους. Η ομοιότητα μεταξύ δύο κόμβων γίνεται με την χρήση του τύπου *Extended Jaccard Coefficient*

$$corr_{i,j} = \frac{b_i(t)b_j(t)}{\|b_i(t)\|_2^2 + \|b_j(t)\|_2^2 - b_i(t)b_j(t)}$$

όπου $b_i(t)$ είναι οι τελευταίες X τιμές του κόμβου

$$\|b_i(t)\|_2^2 = |x_i(t-\Delta t+1)|^2 + \cdots + |x_i(t)|^2$$

(με τιμή κοντά στο 1 θεωρείται ότι είναι σχετικά όμοιες οι μετρήσεις των κόμβων και με τιμή κοντά στο 0 θεωρείται ότι είναι ανόμοιες οι μετρήσεις των κόμβων). Επίσης κάθε κόμβος έχει ένα βαθμό(*rank*) το οποίο δείχνει την αξιοπιστία του κόμβου, και το οποίο εξαρτάται από το πόσους όμοιους γείτονες έχει και από το πόσο αξιόπιστοι είναι οι γείτονές του. Ο αλγόριθμος που παρουσιάζουν είναι επαναληπτικός και έχει τα εξής βήματα:

- Όλοι οι κόμβοι αρχικοποιούνται με βαθμό ($rank = 1$) και υπολογίζουν τις συσχετίσεις με τους γείτονές τους

$$p_{i,j} = \frac{corr_{i,j}}{\sum_{s_k \in \text{γείτονες}(s_i)} corr_{i,k}}$$

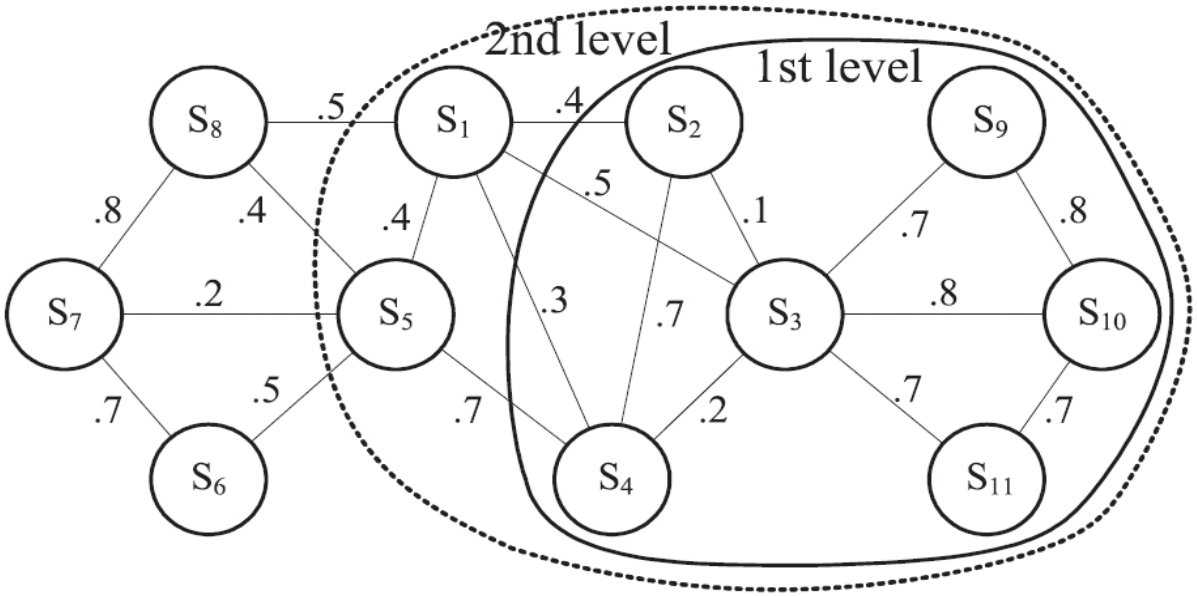
- Στη συνέχεια κάθε κόμβος στέλνει στους γείτονες το βαθμό που είχε υπολογίσει στο προηγούμενο βήμα και υπολογίζει το νέο του

$$rank_i^{(k)} = \sum_{s_j \in \text{γείτονες}(s_i)} rank_j^{(k-1)} \times p_{i,j}$$

- Επαναλαμβάνεται ο αλγόριθμος για δ φόρες που ορίζει ο χρήστης.

- Κάθε κόμβος που κρίνεται ότι περιέχει ακραία τιμή δε συμμετέχει σε ψηφοφορίες για άλλους κόμβους.

Παράδειγμα για τον παραπάνω αλγόριθμο με το σχεδιάγραμμα του δικτύου αισθητήρων και τον πίνακα που παράγει ο αλγόριθμος.



Σχήμα 2.6 Σχεδιάγραμμα του δικτύου αισθητήρων με τα βάρη στις ακμές μεταξύ των γειτόνων

Πίνακας 2.1 Τα αποτελέσματα του αλγορίθμου για το σχήμα 2.6

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}
$k=0$	1	1	1	1	1	1	1	1	1	1	1
$k=1$	1.13	0.59	1.74	1.11	1.33	0.64	1.14	0.89	0.58	1.3	0.54
$k=2$	1.17	0.68	1.43	1.05	1.24	0.77	0.91	1.05	0.86	1.04	0.8

2.3 Κατακερματισμός Τοπικής Ευαισθησίας (Locality-Sensitive Hashing , LSH)

Στην ενότητα αυτή αναφέρουμε σχετικές εργασίες που χρησιμοποιούν τη μέθοδο του κατακερματισμού τοπικής ευαισθησίας(LSH) για την προσεγγιστική αναζήτηση πλησιέστερου γείτονα.

Τη μέθοδο του LSH χρησιμοποιείται και στην εργασία μας για τον εντοπισμό ακραίων μετρήσεων. Στο κεφάλαιο 4 θα δοθεί ανάλυση για την επίλυση του κοντινότερου γείτονα με την βοήθεια του LSH.

2.3.2 Efficient Incremental Near Duplicate Detection based on Locality Sensitive Hashing

Το LSH στη δημοσίευση[10] χρησιμοποιείται για τη λύση του προβλήματος ανίχνευση κοντινών διπλότυπων. Ο στόχος της εργασίας είναι να βρεθούν όλα τα σημεία εκείνα μέσα στο υπάρχον σύνολο, όπου βρίσκονται κοντά στο νέο σημείο. *π.χ. Η εύρεση όλων των εικόνων από ένα ερώτημα ομοιότητας εικόνας σε μία μεγάλη βάση δεδομένων. Η εικόνα μπορεί να αναπαρασταθεί ως ένα διάνυσμα άρα το σύνολο των εικόνων είναι υψηλής διαστατικότητας και μπορεί να χρησιμοποιηθεί ο αλγόριθμος LSH για την μείωση του χρόνου εύρεσης των εικόνων.*

Η εφαρμογή της παρούσας εργασίας είναι η ανίχνευση κοντινών διπλότυπων περιεχόμενων σε πολυμέσα όπως είναι το «flickr»[14] και το «youtube»[15] σε πραγματικό χρόνο. Κάθε φορά που ένας χρήστης φορτώνει μία εικόνα ή ένα βίντεο σε μία ιστοσελίδα θα πρέπει να ανιχνευθούν τα σχεδόν διπλότυπα δηλαδή αυτά που είναι πολύ παρόμοια με εκείνο που φορτώθηκε και να επιστραφούν στο χρήστη σε πραγματικό χρόνο. Ο χρήστης στη συνέχεια θα του δίνεται η δυνατότητα να επιλέξει να συνεχίσει την φόρτωση της εικόνας ή όχι.

Το πρόβλημα αυτής της εφαρμογής είναι ότι εάν χρησιμοποιηθεί μία παραδοσιακή μέθοδο αναζήτησης θα καταλάμβανε μεγάλο χώρο στη μνήμη(λόγο υψηλής διαστατικότητας) και θα ήταν μεγάλος ο χρόνος επεξεργασίας για την εύρεση των σχεδόν διπλότυπων. Άρα ο στόχος της εργασίας ήταν η μείωση κατανάλωση μνήμης και η παροχή γρήγορης αναζήτησης. Η προσέγγιση της εργασίας για να επιτύχει τα παραπάνω βασίζεται στο LSH και ονομάζεται «SimPairLSH». Η κύρια ιδέα του SimPairLSH είναι να επωφεληθούν από ένα ορισμένο αριθμό από υφιστάμενα παρόμοια ζεύγη σημείων. Περισσότερες πληροφορίες για τις τεχνικές που χρησιμοποιεί αυτή η εφαρμογή στη δημοσίευση[10].

ΚΕΦΑΛΑΙΟ 3

STABLE DISTRIBUTIONS

Οι ευσταθής κατανομές(*stable distributions*) είναι μία μεγάλη οικογένεια από τις κατανομές των πιθανοτήτων που επιτρέπουν τη λόξωση(*skew*) και βαριές ουρές(*heavy tails*) έχοντας ενδιαφέρουσες μαθηματικές ιδιότητες. Έχουν προταθεί ως μοντέλα για διάφορα φυσικά και οικονομικά συστήματα έχοντας πολυάριθμες εφαρμογές σε πολλά πεδία. Η υλοποίηση τους μπορεί να γίνει με ποικίλα προγράμματα που υπολογίζουν προσεγγιστικά ευσταθής κατανομές χρησιμοποιώντας τις σε διάφορα πρακτικά προβλήματα που αντιμετωπίζουμε όπως το δικό μας. Στο πεδίο της επιστήμης των υπολογιστών χρησιμοποιείται η τεχνική σκιαγραφώντας(*sketching*) διανύσματα με υψηλή διαστατικότητα(*high dimensional*).

3.1 Εισαγωγή ευσταθών κατανομών

Στην θεωρία πιθανοτήτων μία τυχαία μεταβλητή λέγεται ότι είναι ευσταθής (ή ότι έχει ευσταθή κατανομή), εάν έχει την ιδιότητα ότι ο γραμμικός συνδυασμός δύο ανεξάρτητων αντιγράφων της μεταβλητής έχει την ίδια κατανομή. Αποτέλεσμα αυτής της ιδιότητας είναι ότι αν ο X είναι κανονική τυχαία μεταβλητή, τότε για X_1 και X_2 που είναι ανεξάρτητα αντίγραφα του X και με θετικές σταθερές a και b ισχύει,

$$aX_1 + bX_2 = cX + d, \tag{3,1}$$

για ορισμένα θετικά c και $d \in \Re$.

Ορισμός 3.1 *Μία τυχαία μεταβλητή X είναι ευσταθής ή ευσταθής υπό την ευρεία έννοια εάν για X_1 και X_2 ανεξάρτητα αντίγραφα της μεταβλητής X και με οποιεσδήποτε σταθερές a και b, ισχύει η εξίσωση (3,1) για ορισμένα c και $d \in \Re$. Η τυχαία μεταβλητή είναι αυστηρά ευσταθής εάν η εξίσωση (3,1) ισχύει για $d = 0$ για όλες τις επιλογές a και b. Μία τυχαία μεταβλητή είναι συμμετρικά ευσταθής εάν είναι ευσταθής και συμμετρικά κατανεμημένη γύρω από το 0.*

Ένας προσθετικός κανόνας για τις ανεξάρτητες τυχαίες μεταβλητές λέει ότι ο μέσος όρος του αθροίσματος είναι το άθροισμα των μέσων όρων και η διακύμανση του αθροίσματος είναι το άθροισμα της διακυμάνσεων. Υποθέτουμε $X \sim N(\mu, \sigma^2)$, τότε οι όροι για την αριστερή πλευρά της εξίσωσης (3,1) είναι $X \sim N(a\mu, (a\sigma)^2)$ και $X \sim N(b\mu, (b\sigma)^2)$ αντίστοιχα ενώ στη δεξιά πλευρά της εξίσωσης είναι $X \sim N(c\mu + d, (c\sigma)^2)$. Με τον προσθετικό κανόνα θα πρέπει να έχουμε

$$c^2 = a^2 + b^2 \text{ και}$$

$$d = (a + b - c)\mu.$$

Δύο τυχαίες μεταβλητές X και Y λέγονται ότι είναι της ίδιας κατανομής εφόσον υπάρχουν σταθερές $A > 0$ και $B \in \Re$ με $X = AY + B$.

Παρακάτω παρουσιάζονται παραδείγματα ευσταθών κατανομών όπως η «Gaussian»(κανονική κατανομή), η «Cauchy» κατανομή και η «Levy» κατανομή. Η Gaussian και η Cauchy κατανομή χρησιμοποιούνται στην εργασία μας για την παραγωγή συναρτήσεων στο LSH σύστημα μας.

Παράδειγμα 3.1 Gaussian ή κανονική κατανομή $X \sim N(\mu, \sigma^2)$ εάν έχει συνάρτηση πυκνότητας

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{(x-\mu)^2}{2\sigma^2})}, \quad -\infty \prec x \prec \infty.$$

Η αθροιστική συνάρτηση κατανομή είναι

$$F(x) = P(X \le x) = \Phi((x - \mu)/\sigma),$$

όπου $\Phi(z)$ ισούται με την πιθανότητα να είναι μία κανονική τυχαία μεταβλητή μικρότερη ή ίση του z.

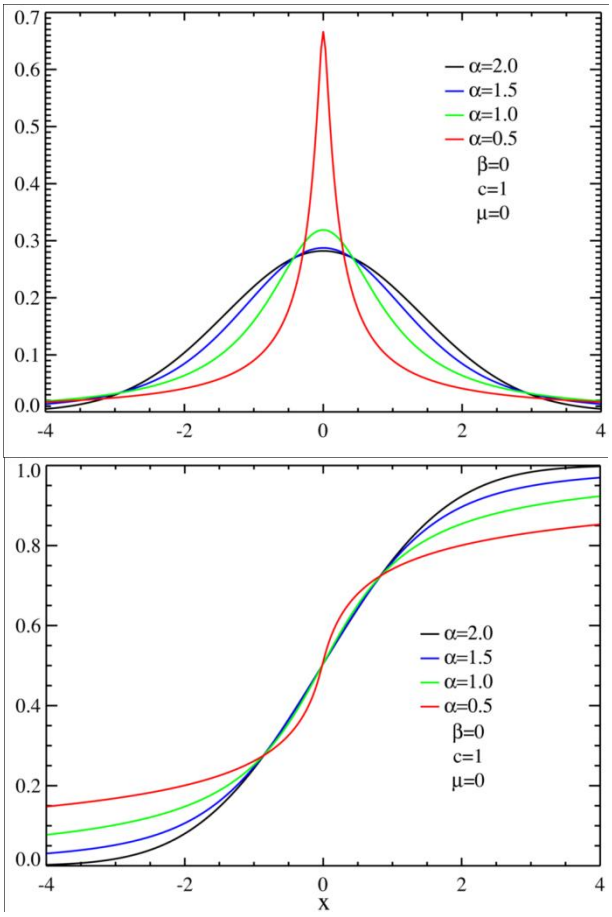
Παράδειγμα 3.2 Cauchy κατανομή $X \sim Cauchy(\gamma, \delta)$ εάν έχει συνάρτηση πυκνότητας

$$f(x) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x - \delta)^2}, \quad -\infty \prec x \prec \infty$$

Αυτές οι κατανομές επίσης καλούνται και Lorentz στη φυσική.

Παράδειγμα 3.3 Levy κατανομή $X \sim Levy(\gamma, \delta)$ εάν έχει συνάρτηση πυκνότητας

$$f(x) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(x-\delta)^{3/2}} \exp(-\frac{\gamma}{2(x-\delta)}),\ \delta<x<\infty$$



Σχήμα 3.1 Αριστερά συνάρτηση πυκνότητας πιθανότητας για διάφορα α και δεξιά αθροιστική συνάρτηση κατανομής για διάφορα α

3.2 P-ευσταθής κατανομές

Οι ευσταθής κατανομές ορίζονται[7] ως τα όρια των κανονικοποιημένων αθροισμάτων από πανομοιότυπες ανεξάρτητες μεταβλητές κατανομών. Η οικογένεια των

ευσταθών κατανομών είναι μεγάλη, μία όμως ευρέως γνωστή ευσταθής κατανομή είναι η Gaussian ή αλλιώς κανονική κατανομή.

Ορισμός 3.2 *P-ευσταθής κατανομή. Μία κατανομή D στο $\Re$ καλείται p-ευσταθής, εάν υφίσταται $p \geq 0$ τέτοιο ώστε για κάθε n πραγματικοί αριθμοί $v_1 \ldots v_n$ και μεταβλητές i.i.d. $X_1 \ldots X_n$ με κατανομή D, η τυχαία μεταβλητή $\sum_i v_i X_i$ έχει όμοια κατανομή με την μεταβλητή*

$$(\sum_i |v_i|^p)^{1/p} X$$

όπου X είναι μία τυχαία μεταβλητή με κατανομή D.

Είναι γνωστό από την δημοσίευση[11] ότι οι ευσταθής κατανομές υπάρχουν για κάθε $p \in (0, 2]$. Ειδικότερα έχουμε δύο p-ευσταθής κατανομές που έχουμε αναφερθεί και παραπάνω:

- Cauchy κατανομή D_C, ορίζεται με συνάρτηση πυκνότητας

$$c(x) = \frac{1}{\pi} \frac{1}{1 + x^2},$$

για $p = 1$

- Gaussian κατανομή D_G, ορίζεται με συνάρτηση πυκνότητα

$$g(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

για $p = 2$

Η κύρια ιδιότητα των p-ευσταθών κατανομών που αναφέρθηκε στον παραπάνω ορισμό μεταφράζεται άμεσα στην τεχνική «σκιαγραφώντας» διανύσματα με υψηλή διαστατικότητα. Η ιδέα είναι να υλοποιηθούν τα παραπάνω και να παραχθεί ένα τυχαίο διάνυσμα **α** με διάσταση d στο οποίο η κάθε είσοδος επιλέγεται τυχαία από μία p-ευσταθής κατανομή. Έτσι δίνοντας ένα διάνυσμα **v** με διάσταση d το εσωτερικό γινόμενο του **α** • **v** είναι μία τυχαία μεταβλητή με την κατανομή της να είναι

$$(\sum_i |v_i|^p)^{1/p} X \quad (\pi.\chi.\ \|v\|_p X)$$

όπου X είναι τυχαία μεταβλητή με p-ευσταθή κατανομή. Μία μικρή συλλογή από τα εσωτερικά γινόμενα που αντιστοιχούν σε διαφορετικά **α** ορίζονται ως το

σκαρίφημα(*sketch*) του διανύσματος v και μπορεί να χρησιμοποιηθεί για την εκτίμηση $\|v\|_p$. Ένα τέτοιο σκαρίφημα παρατηρούμε ότι είναι γραμμικής σύνθεση

$$\mathbf{a \cdot (v_1 - v_2) = a \cdot v_1 - a \cdot v_2}\,.$$

ΚΕΦΑΛΑΙΟ 4

LOCALITY-SENSITIVE HASHING

Η μέθοδος αυτή προτάθηκε από τους Indyk και Motwani[6] και στηρίζεται στην βασική ιδέα: εάν εισαχθούν δεδομένα σ'ένα πίνακα κατακερματισμού(*hash table*), με τέτοιο τρόπο ώστε τα δεδομένα αναζήτησης να είναι πλησίον ενός σημείου αποθηκευμένων δεδομένων στο πίνακα κατακερματισμού, τότε θα έχουν το ίδιο κλειδί κατακερματισμού. Με άλλα λόγια η πιθανότητα «σύγκρουσης» των κλειδιών κατακερματισμού για τα δεδομένα είναι μεγαλύτερη όσο οι τιμές για τα δεδομένα συγκλίνουν. Το LSH συνήθως χρησιμοποιείται σε δεδομένα που έχουν υψηλή διαστατικότητα όπως για αναγνώριση ομοιότητας σε δεδομένα εικόνας, ήχου και βίντεο.

Για την επίλυση του προβλήματός μας που είναι η ανίχνευση ακραίων τιμών σε ΑΔΑ θα χρησιμοποιήσουμε το σχήμα LSH. Στην αρχή θα δοθεί ο ορισμός ενός LSH σχήματος και κάποιους τρόπους κατασκευής του. Στη συνέχεια θα εξηγηθεί το πρόβλημα του κοντινότερου γείτονα, θα αναλύσουμε τη λογική του LSH σχήματος το οποίο θα μας χρησιμεύσει στην επίλυση του κοντινότερου γείτονα.

4.1 Ορισμός του LSH

Μία οικογένεια LSH $\mathcal{F}$ ορίζεται για μετρικό χώρο $\mathcal{M}=(M,d)$ ένα κατώφλι $R>0$ και ένας προσεγγιστικός παράγοντας $c>1$. Μία οικογένεια LSH $\mathcal{F}$ είναι μία οικογένεια από συναρτήσεις $h:M\to S$ που πρέπει να πληρούν τις εξής προϋποθέσεις για οποιαδήποτε δύο σημεία $p,q\in M$, και η συνάρτηση h διαλέγεται τυχαία και ομοιόμορφα από την οικογένεια συναρτήσεων $\mathcal{F}$:

- Εάν d(p,q) ≤ R, τότε h(p)=h(q) (π.χ. p και q συγκρούονται) με πιθανότητα τουλάχιστον P_1.
- Εάν d(p,q) ≥ cR, τότε h(p)=h(q) με πιθανότητα το πολύ P_2

Μία οικογένεια ισχύει όταν $P_1>P_2$. Αυτή η οικογένεια $\mathcal{F}$ καλείται $(R,cR,P_1,P_2)-$ευαισθησίας

4.1.1 Εφαρμογή προβλημάτων LSH

Τα κυριότερα προβλήματα που μπορεί να εφαρμοστεί το LSH για την επίλυσή τους είναι:

i. Εύρεση Κοντινότερου Γείτονα(*Nearest neighbor search , NNS*)
ii. Σχεδόν Διπλότυπη Ανίχνευση(*Near Duplicate Detection , NDD*)
iii. Αναγνώριση Ομοιότητα Εικόνας(*Image Similarity Identification , ISI*)
iv. Αναγνώριση Ομοιότητας Γονιδιακής Έκφρασης(*Gene Expression Similarity Identification , GESI*)
v. Αναγνώριση Ομοιότητας Ήχου(*Audio Similarity Identification , ASI*)

4.1.2 Μέθοδοι κατασκευής οικογένειας LSH

Για την κατασκευή οικογένειας LSH υπάρχουν διάφορες μέθοδοι, οι κυριότερες από τις οποίες είναι οι εξής:

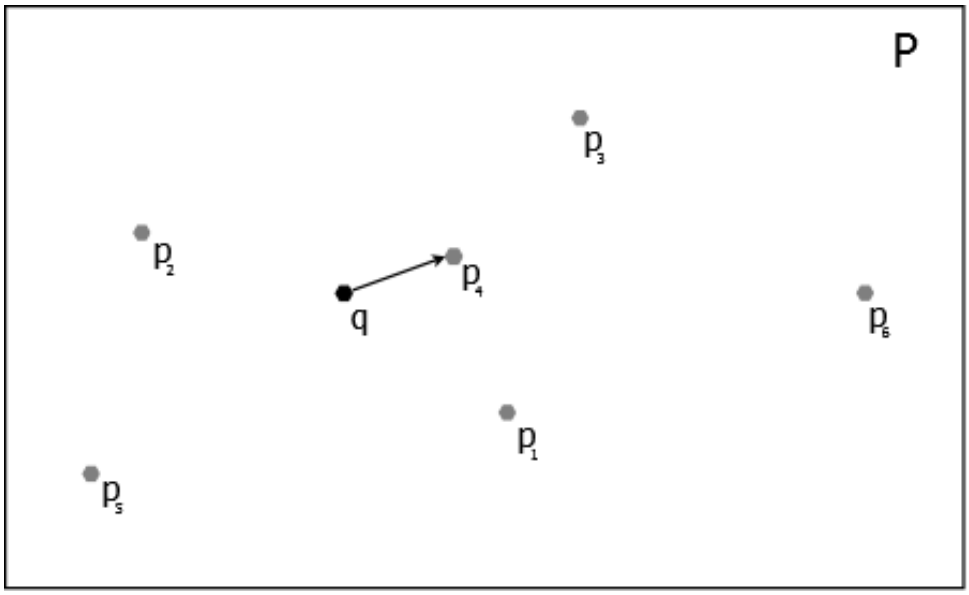
- *Δειγματοληψία «bit» για αποστάσεις «hamming»:*
  Η κατασκευή της οικογένειας γίνεται με δειγματοληψία δυαδικού ψηφίου. Η προσέγγιση αυτή λειτουργεί για αποστάσεις hamming σε d-διαστάσεων διανύσματα $\{0,1\}^d$. Η οικογένεια του συνόλου των προβολών των σημείων σε μία από τις d συντεταγμένη
  $$F = \{h : \{0,1\}^d \to \{0,1\} \mid h(x) = x_i, i = 1...d\}$$
  όπου x_i είναι η ι-οστή συντεταγμένη του x.
- *Ελάχιστο-σοφός για ανεξάρτητες μεταθέσεις(Min-wise independent permutations)*
- *Τυχαία προβολή(Random projection):*
  Η μέθοδος της τυχαίας προβολής είναι σχεδιασμένη έτσι ώστε να προσεγγίζει την απόσταση συνημίτονου δύο διανυσμάτων. Η βασική ιδέα της τεχνικής είναι να επιλέγει ένα τυχαίο υπερεπίπεδο(*hyperplane*) από την αρχή και να χρησιμοποιηθεί το υπερεπίπεδο για την εισαγωγή διανυσμάτων στο πίνακα κατακερματισμού. Δίνοντας εάν διάνυσμα v και ορίζοντας ένα υπερεπίπεδο ως r έχουμε
  $$h(v) = \text{sgn}(v * r).$$
  Δηλαδή $h(v) = \pm 1$ ανάλογα από ποια μεριά του υπερεπιπέδου v βρίσκεται.
- *Ευσταθές Κατανομές:*
  Η συνάρτηση κατακερματισμού
  $$h_{a,b}(v) = \Re^d \to \mathbb{N}$$
  απεικονίζει μία d διάσταση του διανύσματος v μέσα σε ένα σύνολο από πραγματικούς αριθμούς. Κάθε συνάρτηση κατακερματισμού στην οικογένεια κατηγοριοποιείται με μια τυχαία επιλογή του διανύσματος $\boldsymbol{a}$ και μια τιμή b. Όπου $\boldsymbol{a}$ είναι d-διαστάσεων διάνυσμα με καταχωρήσεις που είναι επιλεγμένες ανεξάρτητες από μία ευσταθή κατανομή και b ένας τυχαίος αριθμός επιλεγμένος από μία ομοιόμορφη κατανομή $[0, r]$. Η συνάρτηση hash δίνεται από τον τύπο

$$h_{a,b}(v) = \left\lfloor \frac{a \bullet v + b}{r} \right\rfloor .$$

4.2 Το πρόβλημα του κοντινότερου γείτονα

Το πρόβλημα του κοντινότερου γείτονα που θα αναλύσουμε είναι χρήσιμη για την κατανόηση του προβλήματός μας και έχει ευρεία χρησιμοποίηση στην επιστήμη των υπολογιστών περιλαμβάνοντας τομείς όπως την αναγνώριση προτύπων , αναγνώριση δεδομένων σε πολυμέσα , συμπίεση δεδομένων , την εξόρυξη δεδομένων και σε άλλους ποικίλους τομείς. Επειδή οι τομείς αυτοί χρησιμοποιούν μεγάλο όγκο δεδομένων η υπολογιστική πολυπλοκότητα της αναζήτησης αυτού του προβλήματος έχει μεγάλη σημασία στις μεθόδους που θα χρησιμοποιήσουμε. Ο χρόνος αναζήτησης του προβλήματος είναι πρωτεύον και δευτερεύον είναι η μνήμη που απαιτείται για την εκτέλεση του αλγορίθμου. Στην αρχή θα δοθούν οι ορισμοί για το κοντινότερο γείτονα.

Ορισμός 4.1 *Κοντινότερου Γείτονα δεδομένου ενός συνόλου P από σημεία σε χώρο d-διαστάσεων $\Re^d$, όπου για κάθε ερώτηση q επιστρέφει οποιοδήποτε σημείο p με την μικρότερη απόσταση από το q όπου p ανήκει στο σύνολο P.*



Σχήμα 4.1 Σχηματική αναπαράσταση του ορισμού 4.1.

Το πρόβλημα του κοντινότερου γείτονα δεν έχει καθορίζεται πλήρως στον ορισμό διότι δεν έχει ορισθεί πως υπολογίζεται η απόσταση του σημείου p από το q. Συνήθως μπορούμε να υπολογίσουμε την απόσταση δύο σημείων με την νόρμα της διαφοράς τους

$$\|p-q\|_s = \left(\sum_{i=1}^{d} |p-q|^s \right)^{1/s}$$

υπάρχουν όμως και άλλοι τρόποι υπολογισμού της απόστασης ανάλογα με το πρόβλημα που πρέπει να αντιμετωπίσει.

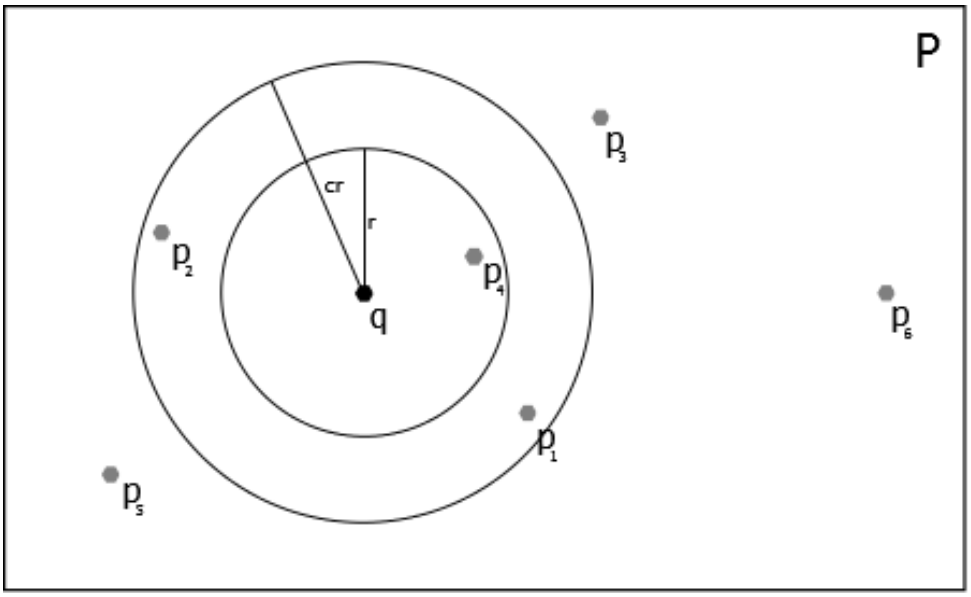
Ως προς το χρόνο επίλυσης του προβλήματος κοντινού γείτονα με την απλή μέθοδο αναζήτηση όλων των σημείων που έχει το σύνολο P και σύγκρισης μεταξύ τους, τότε η αναζήτηση του προβλήματος εκτελείται σε γραμμικό χρόνο $O(n)$ έχοντας μεγάλο όγκο δεδομένων ο χρόνος που θα απαιτείται θα είναι πολύ μεγάλος. Θα πρέπει να βρεθεί ένας υπολογιστικός χρόνος αναζήτησης ώστε να είναι τουλάχιστον λογαριθμικής μορφής. Για να βελτιώσουμε αυτό το χρόνο ένας καλός τρόπος θα είναι να βρίσκουμε τους κοντινούς γείτονες προσεγγιστικά με μία σταθερά c.

Ορισμός 4.2 *C-κατά προσέγγιση κοντινότερου γείτονα δεδομένου ενός συνόλου P από σημεία σε χώρο d-διαστάσεων $\Re^d$, όπου για κάθε ερώτηση q επιστρέφει οποιοδήποτε σημείο σε απόσταση το πολύ c φορές την απόσταση του σημείου q από το p, όπου το p είναι ένα σημείο στο σύνολο P πλησιέστερα στο q.*

Το δικό μας πρόβλημα χρησιμοποιεί τη c-κατά προσέγγιση κοντινότερου γείτονα όμως επιστρέφοντας όλους του κοντινούς γείτονες σε απόσταση cr. Αυτό χρειάζεται για να βρίσκουμε όλα τα διανύσματα που είναι σε μικρή απόσταση.

Ορισμός 4.3 *C-κατά προσέγγιση κοντινών γειτόνων σε απόσταση r δεδομένου ενός συνόλου P από σημεία σε χώρο d-διαστάσεων $\Re^d$, όπου για κάθε ερώτηση q επιστρέφοντας οποιαδήποτε σημεία p του συνόλου P με απόσταση μικρότερη του cr.*

$$\|p-q\|_s < cr$$



Σχήμα 4.2 Σχηματική αναπαράσταση του ορισμού 4.3.

4.2 Επίλυση του κοντινού γείτονα με LSH

Στην ενότητα αυτή θα εξηγηθεί πως συνδυάζεται το πρόβλημα c-κατά προσέγγιση κοντινών γειτόνων με απόσταση R με το σχήμα LSH βασίζοντας στην τυχαιότητα. Παρακάτω δίνονται δύο χρήσιμοι ορισμοί που χρησιμοποιούνται για την αναζήτηση του κοντινού γείτονα μέσω LSH.

Ορισμός 4.4 *Σημείο p είναι R-κοντινός γείτονας του q εάν η απόσταση από το p στο q είναι το πολύ R.*

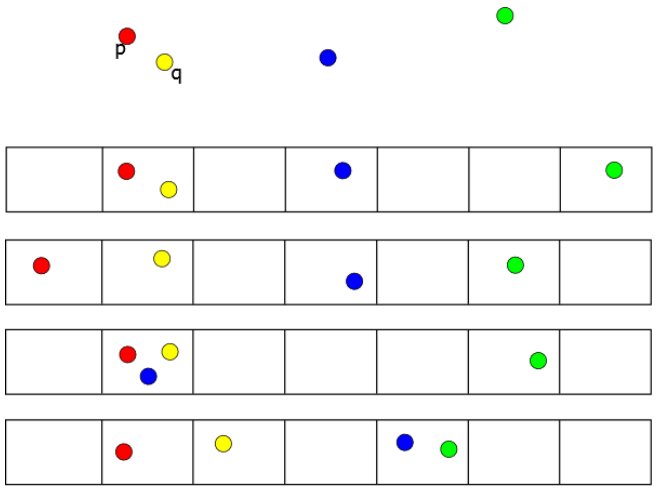
Ορισμός 4.5 *δεδομένου ενός συνόλου P από σημεία σ'ένα d-διάστατο χώρο $\Re^d$ και με παραμέτρους R>0 και δ>0, όπου για κάθε ερώτηση q επιστρέφει με πιθανότητα 1-δ: εάν υπάρχει ένας R-κοντινός γείτονας του q στο P. Αυτό αναφέρεται ως ένας cR-κοντινός γείτονας του q στο P.*

Η βασική ιδέα που χρησιμοποιεί το σχήμα LSH στην επίλυση του προβλήματος είναι ότι κατακερματίζοντας τα σημεία των δεδομένων στους πίνακες κατακερματισμού και χρησιμοποιώντας διαφορετικές συναρτήσεις κατακερματισμού εξασφαλίζεται ότι για κάθε συνάρτηση η πιθανότητα σύγκρουσης είναι πολύ υψηλότερος για σημεία που είναι κοντά το ένα στο άλλο παρά για σημεία που είναι μακριά. Ως αποτέλεσμα αυτού ένα σημείο p μπορεί να καθορίσει τους κοντινούς γείτονες ανατρέχοντας τα σημεία που έχουν κατακερματιστεί στους ίδιους κουβάδες με αυτόν.

Το LSH μπορεί να χρησιμοποιηθεί είτε για την κατά προσέγγιση αναζήτηση κοντινού γείτονα ή για την αναζήτηση απλά των κοντινών γειτόνων. Με βάση τον ορισμό που έχουμε δώσει στην ενότητα 4.1 για το LSH θα το χρησιμοποιήσουμε στο πρόβλημά μας και έχουμε μία οικογένεια H η οποία καλείται (R, cR, P_1, P_2)-ευαισθησίας εάν για κάθε $p, q \in \Re^d$

- Αν $\|p-q\| \le R$ τότε $P_H[h(q)=h(p)] \ge P_1$
- Αν $\|p-q\| \ge cR$ τότε $P_H[h(q)=h(p)] \le P_2$

μία LSH οικογένεια έχει υπόσταση όταν $P_1 > P_2$.



Σχήμα 4.3 Σχηματική αναπαράσταση του LSH. Στο σχήμα παρουσιάζεται ένα παράδειγμα του LSH με 4 σημεία τα οποία αρχικά δίνονται στη χωρική τους τοποθεσία και στη συνέχεια δείχνεται πως θα μπορούσαν να αποθηκεύονται σε ένα LSH σχήμα. Βλέπουμε ότι υπάρχει μεγάλη πιθανότητα κοντινοί γείτονες να είναι και στους ίδιους κουβάδες.

Η LSH οικογένεια χρησιμοποιείται ως εξής. Με δεδομένη οικογένεια H από συναρτήσεις κατακερματισμού με τις παραμέτρους (R, cR, P_1, P_2), ενισχύεται το χάσμα μεταξύ της υψηλής πιθανότητας P_1 και της χαμηλής πιθανότητας P_2 με την συνένωση διαφορετικών συναρτήσεων. Ειδικότερα το k και το L καθορίζονται αργότερα. Επιλέγοντας τυχαία ανεξάρτητα και ομοιόμορφα από την οικογένεια H τις L συναρτήσεις θα έχουμε

$$g_j(q) = (h_{1,j}(q), ..., h_{k,j}(q)) \text{ όπου } h_{t,j} (1 \le t \le k,\ 1 \le j \le L)$$

Τέλος κατά την διάρκεια της προεπεξεργασίας αποθηκεύουμε στο κουβά $g_j(p)$, για $j = 1, ..., L$ για το κάθε σημείο που ανήκει στο P.

4.3 Το LSH σχήμα μας

Το δικό μας σχήμα LSH βασίστηκε στους παραπάνω ορισμούς , ιδέες και στο LSH σχήμα της δημοσίευσης [7]. Ο ορισμός της οικογένειας μας του LSH είναι $H = \{h : S \to U\}$ και καλείται (r_1, r_2, p_1, p_2) -ευαισθησίας για D εάν για κάθε $v, q \in S$

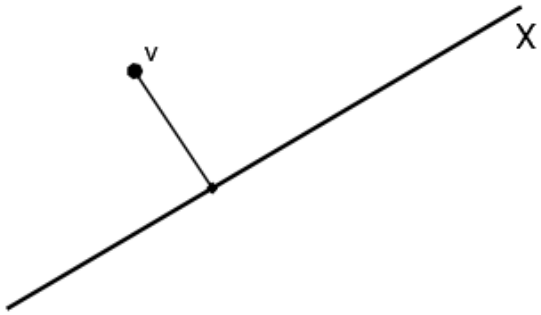
- Εάν $v \in B(q, r_1)$ τότε $\Pr_H[h(q) = h(v)] \geq p_1$,
- Εάν $v \notin B(q, r_2)$ τότε $\Pr_H[h(q) = h(v)] \leq p_2$.

Ισχύει για $p_1 > p_2$ και $r_1 < r_2$. Επιλέγουμε το $r_1 = R$ και $r_2 = cR$. Σ'ένα ερώτημα q η αναζήτηση γίνεται σε όλους τους κουβάδες $g_1(q),...,g_L(q)$ για να βρεθούν οι πιθανοί κοντινοί γείτονες. Έστω ότι έχουμε $v_1,...,v_L$ που είναι σημεία μας τότε για κάθε v_j επιστρέφει ΝΑΙ εάν $v_j \in B(q, r_2)$ ειδάλλως ΟΧΙ.

Για να βρούμε τα βέλτιστα *k* και *L* χρησιμοποιούμαι στην αρχή του προγράμματος μας ένα δείγμα από τις μετρήσεις του κάθε κόμβου. Το σκεφτικό είναι έχοντας ένα τυχαίο δείγμα για κάθε κόμβο, γίνονται ερωτήσεις στο σύνολο των δεδομένων για διάφορα *k* και *L*. Επιλέγονται τα *k* και *L* για τα οποία ο χρόνος απάντησης ερωτήματος είναι. Αναλυτικότερα η διαδικασία περιγράφεται στην ενότητα 5.4.

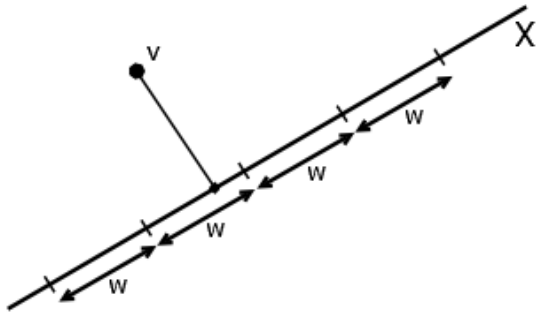
4.2.1 Οι ευσταθείς κατανομές για τη δημιουργία της οικογένειας συναρτήσεων στο LSH σχήμα μας

Σε αυτή την υποενότητα παρουσιάζεται το LSH σχήμα βασισμένο στις p-ευσταθές κατανομές για p=1(Cauchy κατανομή) και p=2(Gaussian κατανομή) για τη δημιουργία της οικογένειας συναρτήσεων. Στο κεφάλαιο 3 εξηγήθηκαν οι ευσταθής κατανομές και δόθηκε και ο ορισμός τους. Για να θυμηθούμε η κύρια ιδιότητα των ευσταθών κατανομών είναι η τεχνική «σκιαγραφώντας» διανύσματα με υψηλή διαστατικότητα. Θα το εκμεταλλευτούμε με τα εσωτερικά γινόμενα $(a \bullet v)$ ώστε να προσδιορίσουμε την τιμή κατακερματισμού για κάθε διάνυσμα v. Διαισθητικά βλέπουμε ότι θα πρέπει η συνάρτηση κατακερματισμού για δύο διανύσματα v_1, v_2 που είναι κοντά (δηλαδή μικρή απόσταση $\|v_1 - v_2\|_p$) να έχει μεγάλη πιθανότητα σύγκρουσης δηλαδή να βρίσκονται στον ίδιο κουβά ή την ίδια τιμή κατακερματισμού, σε αντίθετη περίπτωση εάν έχουν μεγάλη απόσταση μεταξύ τους θα πρέπει να έχουν μικρή πιθανότητα σύγκρουσης. Το εσωτερικό γινόμενο για κάθε διάνυσμα v με το διάνυσμα a(διάνυσμα p-ευσταθής κατανομής) προβάλλεται στη πραγματική γραμμή.



Σχήμα 4.4 Προβολή του εσωτερικού γινομένου $(a \bullet v)$ στη πραγματική γραμμή

Θα τεμαχιστεί η πραγματική γραμμή σε ίσα πλάτη τμήματα w ώστε οι τιμές κατακερματισμού των διανυσμάτων να προβάλλονται μέσα σε αυτά τα τμήματα για να δημιουργήσουμε τη λεγόμενη τοπική ευαισθησία. Αποτέλεσμα αυτού θα είναι ότι το κάθε τμήμα w θα είναι και ένας κουβάς στο πίνακα κατακερματισμού στο LSH σχήμα μας.



Σχήμα 4.5 Προβολή του εσωτερικού γινομένου $(a \bullet v) / w$ στη πραγματική γραμμή

Η κάθε συνάρτηση κατακερματισμού θα πρέπει να αναπαριστά ένα διάνυσμα v με μέγεθος d σε ένα σύνολο από ακέραιους αριθμούς. Κάθε συνάρτηση θα δεικτοδοτείται από ένα διάνυσμα a (διάνυσμα p-ευσταθής κατανομής μεγέθους d), από μία πραγματική τιμή b (επιλέγεται ομοιόμορφα στο διάστημα $[0, w]$) και w το μέγεθος των τμημάτων στην πραγματική γραμμή. Ο τύπος της συνάρτησης κατακερματισμού είναι

$$h_{a,b} = \left\lfloor \frac{(a \bullet v) + b}{w} \right\rfloor$$

Αφού εξηγήθηκε η χρησιμότητα των p-ευσταθών κατανομών για τη δημιουργία της οικογένειας συναρτήσεων θα εξηγηθεί στην ενότητα 5.6 η δημιουργία L οικογένειες στο πρόγραμμά μας.

4.2.2 Πιθανότητα σύγκρουσης δύο διανυσμάτων

Οι πιθανότητες σύγκρουσης δύο διανυσμάτων έχουν υπολογιστεί στη δημοσίευση [7] και εξαρτάται από το βαθμό της p-ευσταθούς κατανομής.

Για p=1(Cauchy) η πιθανότητα σύγκρουσης p_1 και p_2 είναι:

$$p_1 = 2\frac{\tan^{-1}(w)}{\pi} - \frac{1}{\pi(w)}\ln(1+w^2)$$

$$p_2 = 2\frac{\tan^{-1}(w/c)}{\pi} - \frac{1}{\pi(w/c)}\ln(1+(w/c)^2)$$

και για p=2(Gaussian) η πιθανότητα σύγκρουσης p_1 και p_2 είναι:

$$p_1 = 1 - 2norm(-w) - \frac{2}{\sqrt{2\pi w}}(1-e^{r^2/2})$$

$$p_2 = 1 - 2norm(-w/c) - \frac{2}{\sqrt{2\pi w}/c}(1-e^{r^2/2c^2})$$

όπου norm είναι η αθροιστική κατανομή της Gaussian κατανομής.

Ο λόγος των p_1 και p_2 πιθανοτήτων σύγκρουσης πρέπει να είναι:

$$\rho = \frac{\ln(1/p_1)}{\ln(1/p_2)}$$

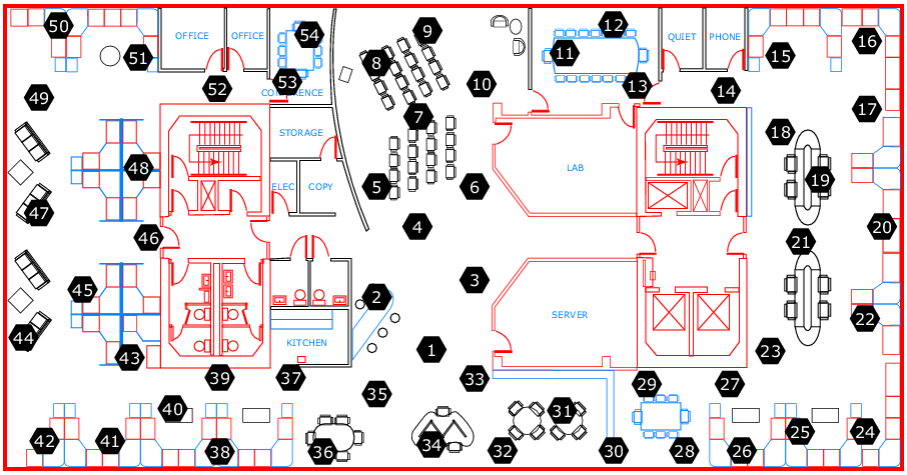
Υπολογίζεται και συγκρίνεται με το 1/c που θεωρείται ο βέλτιστος λόγος.

ΚΕΦΑΛΑΙΟ 5

ΣΧΕΔΙΑΣΜΟΣ & ΥΛΟΠΟΙΗΣΗ

Σε αυτό το κεφάλαιο θα αναλυθεί ο σχεδιασμός και η υλοποίηση του προγράμματος μας. Για την υλοποίηση του αλγορίθμου μας και την προσομοίωση του δικτύου αισθητήρων για να εξάγουμε και να συγκρίνουμε τα αποτελέσματα χρησιμοποιήσαμε το λειτουργικό σύστημα Linux Ubuntu 10.04 και την γλώσσα προγραμματισμού C με τον compiler g++. Το πλεονέκτημα της C είναι ότι είναι γλώσσα υψηλού επιπέδου που ενδείκνυται για προσομοιώσεις και συγκρίσεις αποτελεσμάτων λόγω της υψηλής ταχύτητας εκτελέσεως των προγραμμάτων.

Για την εκτέλεση του προγράμματός μας έχουμε δημιουργήσει τα αρχεία «parameters», «makefile» , «nodes» , «simulation.c» και ένα φάκελο «sensor_data» ο οποίος περιέχει αρχεία με τις μετρήσεις των αισθητήρων. Π.χ. τα δεδομένα που χρησιμοποιήσαμε είναι από πραγματικές μετρήσεις του δικτύου αισθητήρων Intel Berkeley Research Lab την περίοδο 24 Φεβρουαρίου εώς 4 Απριλίου 2004 παίρνοντας μετρήσεις κάθε 31 δευτερόλεπτα. Το εργαστήριο έχει τοποθετημένους 32 αισθητήρες μετρώντας την θερμοκρασία, την υγρασία, την φωτεινότητα και την τάση των τιμών. Τα δεδομένα και επιπλέον πληροφορίες βρίσκονται στο διαδίκτυο [12].



Εικόνα 5.1 Κάτοψη του Intel Research Berkeley με το δίκτυο αισθητήρων

5.1 Περίληψη λειτουργίας προγράμματος

Το πρόγραμμα για να δημιουργήσει το ιεραρχικό ΔΑ και να υλοποιήσει τους αλγόριθμους με τους οποίους θα γίνει σύγκριση για την εξαγωγή των ακραίων τιμών σ'ένα ΔΑ ακολουθεί την εξής διαδικασία:

- Διάβασμα από το αρχείο *parameters* τις παραμέτρους που έχουν οριστεί για τον πρόγραμμα
- Δημιουργία δέντρου των αισθητήρων με βάση το αρχείο nodes το οποίο περιέχει και τις συντεταγμένες των κόμβων
- Επιλογή αλγόριθμου για τον υπολογισμό των ακραίων τιμών (την μέθοδο τη δικιά μας ή την άπληστη μέθοδο)
- Με βάση ενός δείγματος μετρήσεων από τους κόμβους αποφασίζεται για τις βέλτιστες τιμές των παραμέτρων του αλγόριθμού μας
- Δημιουργία των διανυσμάτων ***α*** που ακολουθούν ευσταθές κατανομές όπως η Gaussian ή Cauchy ανάλογα την επιλογή και του διανύσματος ***b*** που ακολουθεί ομοιόμορφη κατανομή
- Διαβάζονται σε κάθε εποχή οι μετρήσεις του κάθε αισθητήρα και εφαρμόζεται ο αλγόριθμός μας ή ο άπληστος αλγόριθμος
- Παράλληλα μετρούνται τα δεδομένα που στέλνονται στο ΔΑ σε bytes

5.2 Παράμετροι προγράμματος

Οι παράμετροι που ορίζουμε στην αρχή του προγράμματός μας έχουν σχέση με την κατασκευή του δέντρου αισθητήρων και τα βασικά χαρακτηριστικά που θέλουμε για τον αλγόριθμο μας. Η επιλογή των τιμών των παραμέτρων γίνονται με βάση τα χαρακτηριστικά του ΑΔΑ που θα έχουμε. *Π.χ. την αξιοπιστία που θέλουμε να έχει ο αλγόριθμός μας, την απόσταση μεταξύ δύο διανυσμάτων για να θεωρούνται ακραίες τιμές, τα μεγέθη των πινάκων που θα έχει ο κάθε αισθητήρας αναλόγως με το μέγεθος μνήμης που έχει, την απόσταση που θέλουμε να έχουν δύο κόμβοι για να θεωρούνται γειτονικοί κλπ.* Παρακάτω εξηγούνται αναλυτικά οι παράμετροι που ορίζονται στην αρχή του προγράμματος.

- *delta* : Ορίζει την πιθανότητα ένα κοντινό διάνυσμα να μην αναφερθεί με πιθανότητα δ. Η επιτυχής πιθανότητα δηλαδή είναι τουλάχιστον 1-δ.
- *R* : Ορίζει τη μέγιστη απόσταση που μπορούν να έχουν τα διανύσματα μετρήσεων δύο διαφορετικών διανυσμάτων ειδάλλως είναι ακραία τιμή

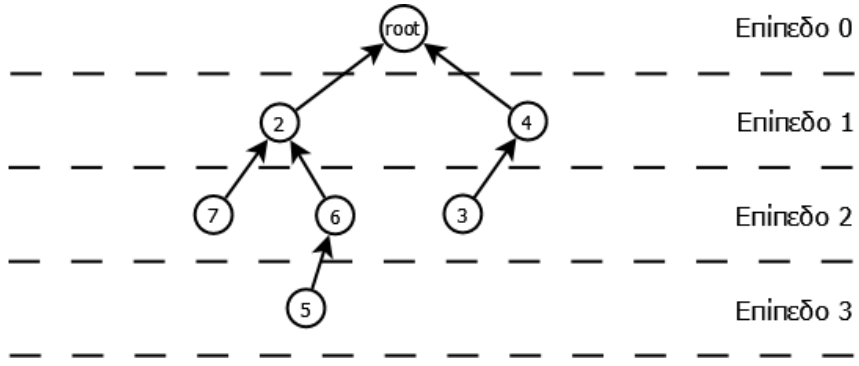
- *minsupport* : Ορίζει τον ελάχιστο αριθμό διανυσμάτων στο πίνακα μετρήσεων ενός κόμβου που πρέπει να βρεθούν όμοιοι για να μην θεωρηθεί το καινούριο διάνυσμα ακραία μέτρηση
- *size_batch* : Ορίζει τον αριθμό των μετρήσεων που περιέχει ένα διάνυσμα μετρήσεων
- *p* : Ορίζει το βαθμό της ευσταθούς κατανομής που διαλέγουμε. Για p=1 είναι η Cauchy ευσταθής κατανομή και για p=2 είναι η Gaussian ευσταθής κατανομή
- *numOfBucketsPerHashTable* : Ορίζει τον αριθμό των κουβάδων που περιέχει ο κάθε πίνακας κατακερματισμού.
- *numOfVectorsInDataSet* : Ορίζει τον αριθμό των διανυσμάτων που μπορούν να αποθηκευτούν σ'ένα πίνακα ενός κόμβου (Διαλέγεται ανάλογα την αποθηκευτική δυνατότητα που έχει ο αισθητήρας).
- *probabilitySentParentNode* : Ορίζεται η πιθανότητα ένα διάνυσμα ενός παιδιού να αποθηκευτεί στο κόμβο γονέα του.
- *dynamicDataSet* : Ορίζουμε εάν θα χρησιμοποιηθεί δυναμικό μέγεθος στο *dataSet* στους κόμβους. Με *dynamicDataSet*=1 χρησιμοποιείται η τεχνική αυτή.
- *factorDynamicDataSet* : Ορίζουμε το ποσοστό που αυξάνεται ο δυναμικός πίνακας.
- *broadcastDist* : Ορίζει τη μέγιστη τοπολογική απόσταση που μπορούν να έχουν δύο αισθητήρες για να θεωρούνται γειτονικοί. Ο κάθε αισθητήρας έχει συντεταγμένες στον άξονα x και y και η τοπολογική απόσταση δύο αισθητήρων ισούται με

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- *read_epoch* : Ορίζονται οι μετρήσεις που διαβάζονται σε μία εποχή.
- *size_int* : Ορίζει το μέγεθος ενός ακέραιου αριθμού σε bytes. Χρησιμοποιείται στον υπολογισμό των bytes που στέλνονται στο δίκτυο των αισθητήρων.
- *size_float* : Ορίζει το μέγεθος ενός πραγματικού αριθμού σε bytes. Χρησιμοποιείται στον υπολογισμό των bytes που στέλνονται στο δίκτυο των αισθητήρων .
- *withOrNotSlidingWindow* : Ορίζει την χρησιμοποίηση ή όχι της μεθόδου ολισθημένο παράθυρο(*sliding window*). Για την τιμή 1 χρησιμοποιείται η μέθοδος και για την τιμή 0 δεν χρησιμοποιείται.
- *slidingWindow_w* : Ορίζει το μέγεθος του παραθύρου που χρησιμοποιούμε στη μέθοδο ολισθημένου παράθυρου.
- *method* : Με την παράμετρο method επιλέγουμε ένα από τους δύο αλγόριθμους που έχουμε υλοποιήσει. Για *method*=1 εκτελείται ο δικός μας αλγόριθμος και για *method*=2 εκτελείται η άπληστη μέθοδος
- *run_epochs* : Ορίζει τις εποχές που θα τρέξουν οι αλγόριθμοι

5.3 Δημιουργία δέντρου αισθητήρων

Στην ενότητα αυτή περιγράφεται η κατασκευή ενός δέντρου με βάσεις τις παραμέτρους που έχουν οριστεί για το πρόγραμμα στην αρχή. Δύο βασικά χαρακτηριστικά του δέντρου ως δομή είναι ότι οι κόμβοι του δέντρου θα είναι όσοι είναι οι αισθητήρες του δικτύου με ρίζα δέντρου το σταθμό βάσης και ότι κάθε κόμβος έχει ένα πατέρα. Σαν παράμετρο για την κατασκευή του δέντρου χρησιμοποιείται η μεταβλητή *broadcastDist* όπου ορίζει την απόσταση ανάμεσα σε δύο αισθητήρες για να θεωρούνται γειτονικοί. *Π.χ. Οι αισθητήρες που είναι γειτονικοί με το κόμβο σταθμό βάσης θα είναι παιδιά του και αυτό θα ισχύει ούτε καθεξής.*



Σχήμα 5.1 Δομή του δέντρου αισθητήρων

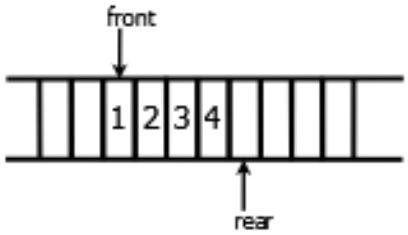
Για την κατασκευή του δέντρου και για την μετέπειτα λειτουργία των μεθόδων ο κάθε κόμβος χρειάζεται να κρατάει την εξής πληροφορίες:

- Αριθμό ταυτότητας(*id*) του κόμβου το οποίο είναι μοναδικό
- Εάν είναι σταθμός βάσης(παράμετρος η οποία βρίσκεται στο αρχείο *nodes*) άρα γίνεται αυτόματα και ρίζα του δέντρου
- Αρχικοποίηση με βάση το αρχείο nodes των συντεταγμένων του κάθε αισθητήρα στον άξονα x και y
- Αποθήκευση του επιπέδου στο οποίο βρίσκεται ο κόμβος στο δέντρο, τον αριθμό ταυτότητας του πατέρα του, τον αριθμό των παιδιών που έχει και τα id των παιδιών του επίσης.
- Δείκτη ο οποίος δείχνει στην ρίζα του δέντρου για την γρήγορη προσπέλαση του δέντρου
- Εάν είναι ο κόμβος φύλλο του δέντρου
- Μετρητή των κανονικών διανυσμάτων που έχει μετρήσει ο κόμβος
- Την πιθανότητα εισαγωγής μίας νέας τιμής στο πίνακα με τις μετρήσεις του κόμβου

- Εάν είναι ακραία η τιμή που μέτρησε ο αισθητήρας στην τρέχουσα εποχή
- Και τέλος τα δεδομένα που κρατάει ο κόμβος για τη λειτουργία των μεθόδων όπως είναι η προσωρινή μνήμη(*buffer*) για την αποθήκευση των τελευταίων N μετρήσεων , οι πίνακες κατακερματισμού, ο πίνακας όπου αποθηκεύονται οι τιμές των διανυσμάτων, μία λίστα με ακραίες μετρήσεις που έχουν θεωρηθεί από τους κόμβους παιδιά του (σε κάθε εποχή σβήνονται οι προηγούμενες).

5.3.1 Μεθοδολογία κατασκευής δέντρου

Με τη βοήθεια της δομής δεδομένων ουράς(*FIFO*) κατασκευάζουμε το δέντρο ως δομή. Η λογική της δομής ουράς είναι το πρώτο στοιχείο που εισάγεται στη δομή θα είναι και το πρώτο που θα αφαιρεθεί. Χρησιμοποιώντας την ιδιότητα ότι όταν ένα στοιχείο εισαχθεί, θα πρέπει να διαγραφούν όλα τα στοιχεία που είχαν εισαχθεί νωρίτερα για να μπορέσει να κληθεί το νέο στοιχείο (ως στοιχείο θα έχουμε τον κόμβο που θα εισαχθεί στο δέντρο). Η δομή μας θα περιέχει δύο δείκτες τον μπροστινό που δείχνει στην αρχή της ουράς και το πίσω που θα δείχνει το τελευταίο στοιχείο.



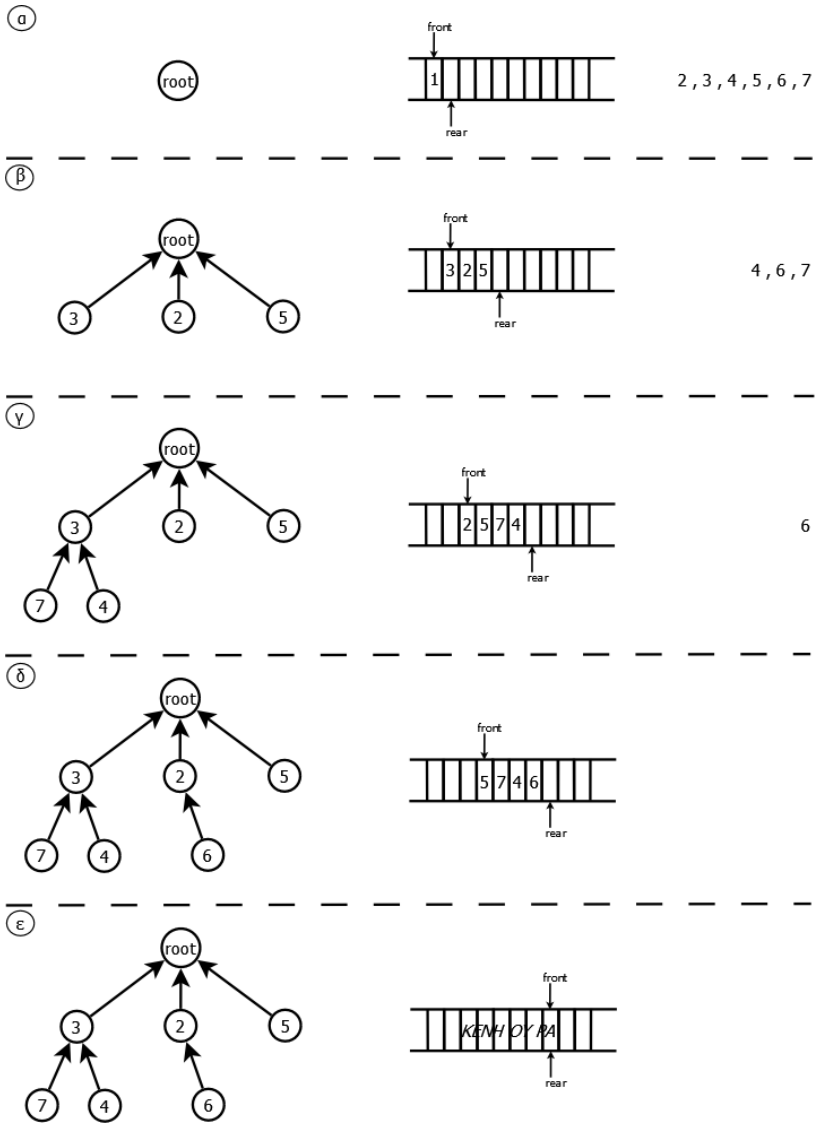
Σχήμα 5.2 Δομή ουράς(FIFO)

Ο αλγόριθμος για την κατασκευή του δέντρου ξεκινάει με την εύρεση του κόμβου σταθμού βάσης προσθέτοντάς την στο τέλος της κενής ουράς. Στη συνέχεια η διαδικασία που περιγράφεται θα είναι επαναλαμβανόμενη μέχρις ότου δεν υπάρχει άλλος κόμβος που δεν ανήκει στο δέντρο(δεν έχει πατέρα).

- Εξαγωγή κόμβου που βρίσκεται στην αρχή της ουράς
- Υπολογισμός απόστασης του κάθε κόμβου που δεν έχουν πατέρα με τον κόμβο που εξάγαμε. Εάν η απόσταση είναι κάτω από το κατώφλι που έχει οριστεί στις παραμέτρους(*broadcastDist*) γίνεται παιδί του και προσθέτεται στο τέλος της ουράς με επίπεδο συν παρότι είναι ο κόμβος-πατέρας.
- Επαναλαμβάνεται το πρώτο βήμα ως η ουρά μείνει άδεια

Παράδειγμα 5.1 Έχουμε 7 αισθητήρες και το κατώφλι απόστασης για να θεωρούνται γείτονες δύο αισθητήρες θα έχει ως αποτέλεσμα ο σταθμός βάσης να έχει γείτονες τους αισθητήρες 3 , 2 και 5. Ο αισθητήρας 3 έχει γείτονες 7 και 4 και ο αισθητήρας 2 έχει

γείτονα τον 6. Ακολουθεί ένα σχηματικό διάγραμμα με την διαδικασία κατασκευής του δέντρου αυτού.



Σχήμα 5.3 Παρουσιάζεται στο σχεδιάγραμμα βήμα βήμα η κατασκευή του δέντρου του παραδείγματος 5.1. Στο σχήμα α) προστίθεται ο σταθμό βάσης στην ουρά μας ο κόμβος 1 και προστίθεται στο δέντρο ως ο κόμβος-ρίζα του δέντρου. Στο β) Αφαιρούμε το κόμβο 1 από την ουρά και βρίσκουμε τα παιδιά του προσθέτοντάς τα παράλληλα στην ουρά και στο δέντρο. Στο γ) Αντίστοιχα με τον κόμβο ένα γίνεται και ο κόμβος 3. Στο δ) Αντίστοιχα και ο κόμβος 2. Στο ε) Όλοι οι κόμβοι που είχανε μείνει στην ουρά δεν έχουν άλλους γείτονες άρα αδειάζει η ουρά και τελειώνει η κατασκευή του δέντρου

5.4 Εύρεση βέλτιστων τιμών στις παραμέτρους του αλγορίθμου μας

Για την βέλτιστη απόδοση της μεθόδους μας ψάχνουμε να βρούμε τα βέλτιστα *k* και *L* ώστε να έχουμε το ελάχιστο κόστος-χρόνος επεξεργασίας (στην παράμετρο *k* και *L* έχουμε αναφερθεί στο κεφάλαιο 4). Στο κόστος το ενδιαφέρον παρουσιάζεται στο χρόνο επεξεργασίας και στην περιορισμένη επικοινωνία μεταξύ των κόμβων που απαιτεί η μέθοδος μας λόγου ότι ο απώτερος στόχος είναι η μείωση της κατανάλωσης ενέργειας στο ΑΔΑ η οποία δεν είναι απεριόριστη. Η μνήμη θεωρείται φθηνή και αρκετή για τα δεδομένα που έχουμε και δεν περιλαμβάνεται υπολογισμό του κόστους. Δοκίμασε διάφορους τρόπους για την απόφαση των βέλτιστων παραμέτρων άλλα δε λειτούργησαν καλά. Έτσι βασιστήκαμε στο τρόπο που χρησιμοποιεί η εργασία[13]. Παρακάτω αναλύεται το σκεφτικό για την βέλτιστη απόφαση παραμέτρων.

Σε συνάρτηση του *k* και του *L* εξαρτάται η αξιοπιστία του σχήματος μας(δηλαδή η παράμετρος που ορίζουμε στην αρχή δ) και ο χρόνος απάντησης ερωτήματος. Για να βρεθεί η σύνδεση που έχουν οι παράμετροι του σχήματος *k* και *L* με το *δ* χρησιμοποιούνται οι ιδιότητες ενός LSH σχήματος που έχουν εξηγηθεί στο κεφάλαιο 4. Έτσι ο τύπος σύνδεσης των παραμέτρων *k,L,δ* από την εργασία[13] είναι

$$L \geq \frac{\log \delta}{\log(1-p_1^k)}$$

Για *k* και *L* σταθερά ο χρόνος απάντησης ερωτήματος T_a μπορεί να διασπασθεί σε δύο όρους

$$T_a = T_g + T_c$$

Ο πρώτος όρος $T_g = O(dkL)$ είναι ο χρόνος που χρειάζεται για τον υπολογισμό των *L* συναρτήσεων g_i για το ερώτημα q καθώς και η επιστροφή των κουβάδων $g_i(q)$ από το πίνακα κατακερματισμού. Ο δεύτερος όρος $T_c = O(d \bullet \#collisions)$ είναι ο χρόνος που χρειάζεται για τον υπολογισμό της απόστασης όλων των σημείων τα οποία βρίσκονται στους κουβάδες $g_i(q)$ με το ερώτημα *q*, ουσιαστικά είναι ο αριθμός των συγκρούσεων που έχουμε στους κουβάδες. Ο τύπος που χρησιμοποιούμαι για την εύρεση της αναμενόμενης τιμής των συγκρούσεων σ'ένα ερώτημα q είναι

$$E[\#collisions] = L \bullet \sum_{v \in P} p^k(\|q-v\|)$$

Παρατηρείται ότι αυξάνοντας το *k* μειώνεται ο χρόνος T_c και ο χρόνος T_g αυξάνεται. Το τελευταίο φαίνεται εύκολα. Το T_c όμως μειώνεται για το λόγο ότι αυξάνοντας το *k* μεγαλώνει το χάσμα στις πιθανότητες που υπάρχουν μεταξύ των μακρινών και κοντινών

σημείων μειώνοντας έτσι την πιθανότητα να έχουν μπει στον ίδιο κουβά δύο σημεία. Τέλος θα συνυπολογιστεί για την εύρεση των βέλτιστων k και L ότι διαφορετικά ερωτήματα μπορούν να βγάλουν διαφορετικά βέλτιστα k και L.

Αναλυτικότερα η διαδικασία που ακολουθείται για την εύρεση των βέλτιστων παραμέτρων στο σχήμα μας χρησιμοποιώντας τα παραπάνω συμπεράσματα είναι:

- αρχικά θέτοντας $k=16$ και $L=\frac{\log\delta}{\log(1-p_1^k)}$ βρίσκονται οι πραγματικοί μέσοι χρόνοι T_g και T_c για διάφορα ερωτήματα από ένα δείγμα του συνόλου δεδομένων
- στη συνέχεια για k από 2 μέχρι 100(ουσιαστικά ο υπολογιστικός χρόνος απάντησης ερωτήματος ανεβαίνει πολύ για $k>100$), υπολογίζεται το L από τον ίδιο τύπο με παραπάνω , υπολογίζεται το T_g ως $L \bullet k \bullet$(πραγματικό μέσο χρόνο T_g) , εκτιμάται ο αριθμός των συγκρούσεων που υπάρχουν από ένα δείγμα ερωτημάτων και τέλος υπολογίζεται το T_c ως

  $$(\text{εκτιμώμενος αριθμός συγκρούσεων}) \bullet (\text{πραγματικό μέσο χρόνο } T_c)$$

- τέλος το μικρότερο άθροισμα $T_g + T_c$ είναι ο βέλτιστος χρόνος άρα και οι βέλτιστοι παράμετροι k και L.

Ο κώδικας που χρησιμοποιήθηκε για την παραπάνω διαδικασία είναι από την εργασία E2LSH[13].

5.5 Δημιουργία διανυσμάτων με ευσταθής κατανομές

Όπως έχει αναφερθεί στην υποενότητα 4.3.1 οι ευσταθείς κατανομές χρησιμεύουν στην κατασκευή συναρτήσεων σε μια οικογένεια LSH. Με βάση τη θεωρία που έχουμε αναλύσει η συνάρτηση κατακερματισμού μας είναι

$$h\ (v)=\left\lfloor \frac{a \bullet v+b}{w} \right\rfloor$$

και πρέπει να παραχθούν τα διανύσματα **α** και **b** για την κατασκευή συναρτήσεων κατακερματισμού**.** Παράγονται $L*k$ διανύσματα **α** μεγέθους d που έχουν ευσταθή κατανομή p=1 ή p=2 και τυχαίο μέσο όρο [0,100]. Το **β** είναι $L*k$ τιμές από ομοιόμορφη κατανομή [0,w].

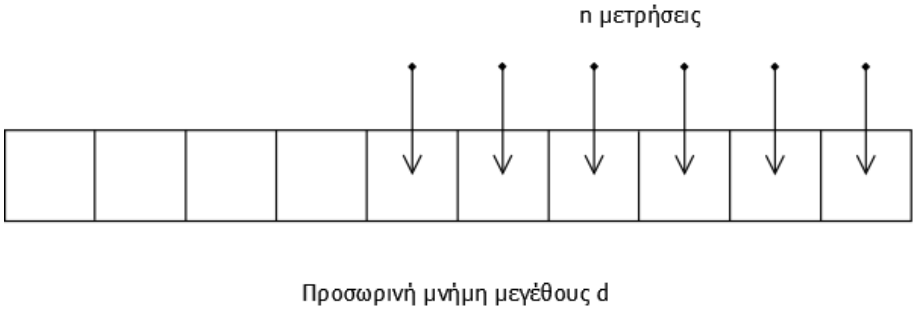
Για την κατασκευή οικογένειας συναρτήσεων χρειάζονται k συναρτήσεις κατακερματισμού. Έτσι η κάθε οικογένεια επιλέγει τυχαία k διανύσματα **α** και τυχαία k τιμές **b**.

Το ***v*** είναι το διάνυσμα μετρήσεων που θα εισαχθεί στο πίνακα κατακερματισμού.

5.6 Λειτουργία αλγορίθμου μας

Η αλγόριθμός μας χρησιμοποιεί το LSH σχήμα όπως έχει αναλυθεί στο κεφάλαιο 4 και συγκεκριμένα στην ενότητα 4.3 για την εύρεση ακραίων τιμών που μπορεί να εμφανίσουν οι κόμβοι. Αφού έχει κατασκευαστεί το δέντρο με τους αισθητήρες έχουν βρεθεί οι βέλτιστοι παράμετροι και έχουν παραχθεί οι L οικογένειες g_i συναρτήσεων όπως έχει εξηγηθεί στις προηγούμενες ενότητες θα περιγραφεί σε αυτή την ενότητα η διαδικασία ανίχνευσης ακραίων τιμών.

Παρακάτω δίνονται κάποιες πληροφορίες για το ΔΑ μας. Ο κάθε κόμβος κρατάει το δικό του πίνακα μετρήσεων μεγέθους *numOfVectorsInDataSet* διανύσματα και L πίνακες κατακερματισμού όσες είναι και οι οικογένειες συναρτήσεων με τον κάθε πίνακα να έχει *numOfBucketsPerHashTable* κουβάδες. Το LSH σχήμα μας έχει κοινές L οικογένειες συναρτήσεων για όλους τους κόμβους. Τέλος σε κάθε εποχή οι μετρήσεις σ'ένα διάνυσμα ολισθαίνουν κατά n ενώ στις τελευταίες n θέσεις του διανύσματος αποθηκεύονται οι n νέες μετρήσεις.



Σχήμα 5.4 Διάβασμα n μετρήσεων και προσθήκη των μετρήσεων στην προσωρινή μνήμη του κόμβου.

Η σάρωση για την εύρεση ακραίων τιμών σε κόμβους σε κάθε εποχή ξεκινάει από το υψηλότερο επίπεδο που είναι τα φύλλα του δέντρου και προχωράει προς τα πάνω επίπεδο επίπεδο έως ότου φτάσει στη ρίζα του δέντρου(σταθμός βάσης) εμφανίζοντας τα αποτελέσματα. Περιγραφή της μεθόδους μας για κάθε εποχή:

- Ξεκινάει από το υψηλότερο επίπεδο που είναι τα φύλλα.
- Έλεγχος εάν είναι ακραίο το διάνυσμα μετρήσεων στο κόμβο. Για να θεωρηθεί ένα διάνυσμα ακραίο θα πρέπει να έχει λιγότερους υποστηρικτές διανυσμάτων από την παράμετρο minsupport που έχει ορισθεί από την αρχή του προγράμματος. Το κατώφλι υποστηρικτών για κάθε κόμβο είναι διαφορετικό γιατί εξαρτάται από το μέγεθος του πίνακα μετρήσεων που έχει και τις φυσιολογικές μετρήσεις που έχουν περάσει.

$$\min SupportNode = \left\lceil \min Support \frac{numOfVectorsInDataSet}{counterNormalVectors} \right\rceil$$

Υποστηρικτής σ΄ένα διάνυσμα θεωρείται όταν βρίσκεται στο ίδιο κουβά, προέρχεται από την ίδια συνάρτηση g_i και η νόρμα της απόστασης των δύο μετρήσεων βαθμού *p* είναι μικρότερη από R. Εάν δεν έχουν βρεθεί επαρκείς υποστηρικτές ψάχνει και στη λίστα των ακραίων διανυσμάτων του κόμβου που έχουν γραφτεί από κόμβους που είναι παιδιά του.

- Στην περίπτωση το διάνυσμα δεν είναι ακραίο θα προστεθεί *+1* στο μετρητή των φυσιολογικών τιμών του κόμβου και θα εκχωρηθεί στο πίνακα διανυσμάτων με έναν από τους δύο τρόπους. (α' τρόπος ολισθημένο παράθυρο κατά w και β' τρόπος με πιθανότητα εισαγωγής στο πίνακα διανυσμάτων)

  α' τρόπος: Χρησιμοποιείται η ιδέα του ολισθημένου παραθύρου όπως στη δημοσίευση[3] έχοντας τον πίνακα που αποθηκεύουμε τα διανύσματα μετρήσεων μεγέθους numOfVectorsInDataSet και ένα παράθυρο μεγέθους *w* το οποίο ολισθαίνει στο χρόνο. Το μέγεθος του παραθύρου είναι τέτοιο ώστε $w \geq numOfVectorsInDataSet$. Δηλαδή θα διαλέγεται τυχαία ένα δείγμα διανυσμάτων numOfVectorInDataSet στο παράθυρο *w* (το κλειδί είναι μοναδικό).

  β' τρόπος: θα εκχωρηθεί στο πίνακα διανυσμάτων σβήνοντας τυχαία ένα παλαιότερο διάνυσμα με πιθανότητα

  $$\frac{\text{(Ο αριθμός των φυσιολογικών τιμών του κόμβου)}}{\text{(Το μέγεθος του πίνακα διανυσμάτων κόμβου)}}$$

- Στην περίπτωση που το διάνυσμα είναι ακραίο εκχωρείται στη λίστα με τα ακραία διανύσματα του κόμβου. Μια ειδική περίπτωση είναι οι κόμβοι φύλλα για να έχουν καλύτερη πληροφορία στα διανύσματα τους που κρατάνε προσθέτουν στο πίνακα και τα διανύσματα που έχουν θεωρηθεί ως ακραία(χωρίς πιθανότητα).
- Στη συνέχεια με την ίδια διαδικασία ελέγχεται ένα ένα τα διανύσματα στη λίστα των ακραίων τιμών που έχουν σταλθεί από το υποδέντρο του και όσα παραμείνουν να θεωρούνται ακραίες τιμές από το κόμβο αυτό και να στέλνονται στο κόμβο πατέρα του.
- Τέλος έχει τεθεί η πιθανότητα *probabilitySentParentNode* η οποία εκφράζει την πιθανότητα να σταλθεί ένα διάνυσμα στο κόμβο πατέρα του ανεξάρτητα εάν είναι ακραίο διάνυσμα. Αυτό γίνεται για να έχει ένας κόμβος καλύτερη πληροφορία για το υποδέντρο του. Επειδή όπως γίνεται αντιληπτό ένας κόμβος που έχει υποδέντρο θα χρειάζεται μεγαλύτερο *dataset* από ένα κόμβο που είναι φύλλο φτιάχτηκε και η παράμετρος του δυναμικού *dataset* η οποία θα εξαρτάται από τον αριθμό των κόμβων που υπάρχουν στο υποδέντρο του κόμβου.

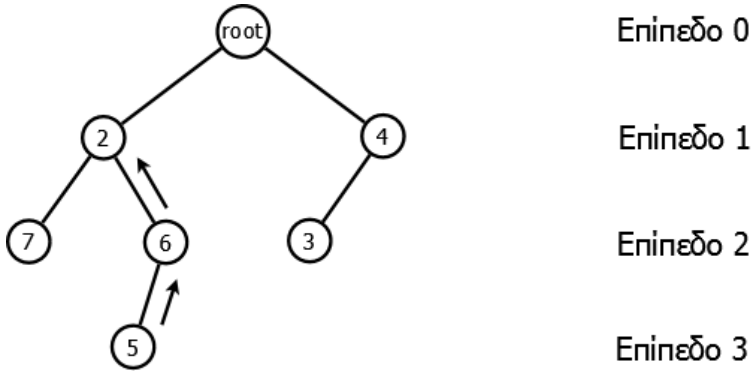
$$sizeDataSet += \left\lceil factorDynamicDataSet * sizeDataSet * numOfNodesInSubTree \right\rceil$$

- Τα βήματα από το πρώτο μέχρι το έκτο επαναλαμβάνονται μέχρι να φτάσει στην ρίζα. Τα διανύσματα που θα μείνουν στη λίστα των ακραίων διανυσμάτων του σταθμού βάσης είναι και οι ακραίες τιμές που θεωρεί για την τρέχων εποχή στο ΑΔΑ.

5.6.1 Παράδειγμα εύρεσης ακραίων τιμών στο ιεραρχικό δέντρο

Χρησιμοποιώντας το δέντρο αισθητήρων του σχήματος 5.1 (ενότητας 5.2) θα δοθεί το παρακάτω παράδειγμα για την εύρεση ακραίων τιμών μέσω της ιεραρχικής δομής που έχουμε ορίσει.

Παράδειγμα 5.2 Ο κάθε κόμβος έχει πίνακα διανυσμάτων μεγέθους N=1000(για απλότητα του παραδείγματος δεν χρησιμοποιείται η πιθανότητα εισαγωγής στο πατέρα-κόμβο και το δυναμικό dataset). Ορίζουμε ως M_i τις φυσιολογικές τιμές που έχουν περάσει από τον ι-οστό κόμβο. Κάνοντας την εξής παραδοχή ότι το διάνυσμα του κόμβου *5* θεωρείται ακραίο για τους κόμβους *5* και *6*, και ότι οι υπόλοιποι κόμβοι δεν έχουν διανύσματα που να θεωρούνται ακραία θα γίνει η εξής διαδικασία σε μια εποχή:

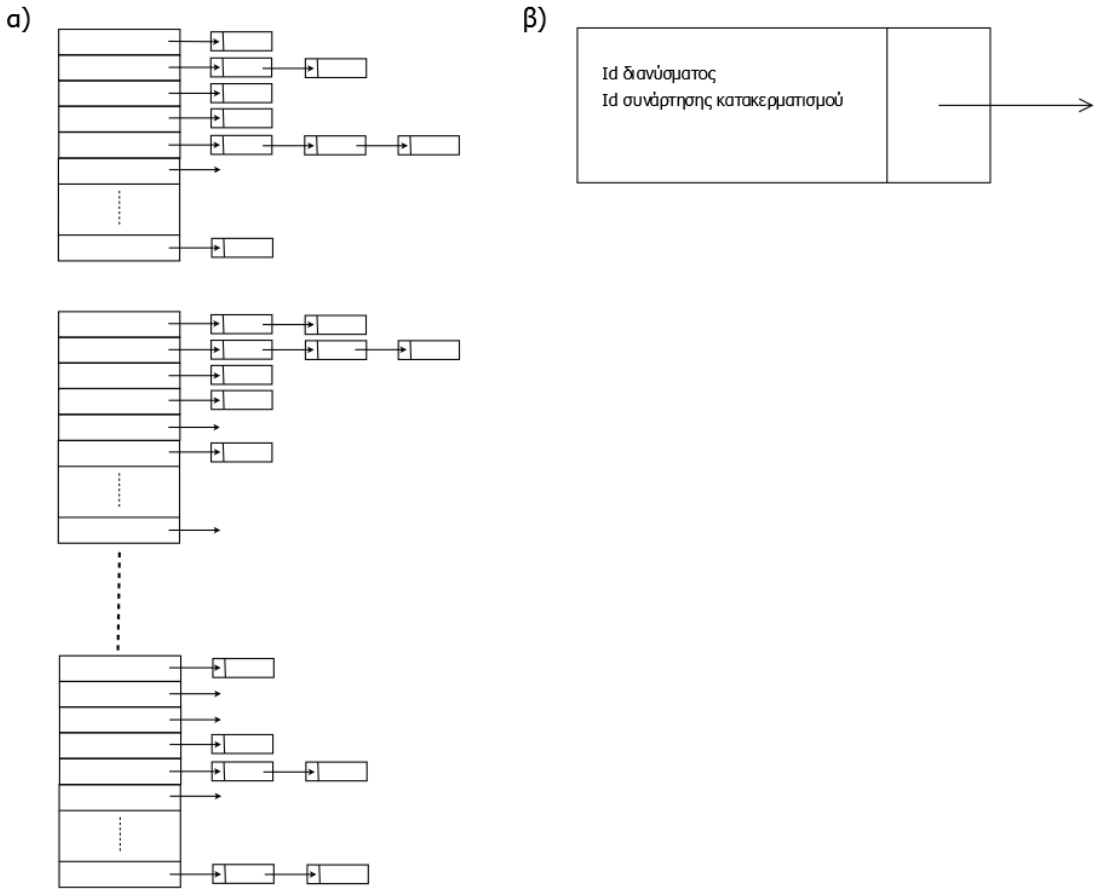


- Όλοι οι κόμβοι προσθέτουν στην προσωρινή μνήμη τους τις καινούριες μετρήσεις τους ολισθαίνοντας τις παλιές τιμές κατά όσες είναι αυτές.
- Η διαδικασία αρχίζει από το υψηλότερο επίπεδο που βρίσκεται ο κόμβος 5. Ο κόμβος έχει ακραία μέτρηση η οποία προστίθεται στη λίστα του με τις ακραίες τιμές. Στη συνέχεια στέλνεται στο παραπάνω επίπεδο στο πατέρα
- Ο κόμβος έξι ελέγχει τη δικιά του μέτρηση και δεν είναι ακραία μέτρηση και έτσι προστίθεται στο πίνακα του με πιθανότητα N/M_6. Ύστερα ελέγχει τη λίστα με τις ακραίες μετρήσεις από τα παιδιά του και βγάζει το διάνυσμα του *5* ακραίο προσθέτοντας το στη λίστα των ακραίων διανυσμάτων του στέλνοντάς το αντίστοιχα στο κόμβο δύο
- Τέλος ο κόμβος *2* ελέγχει το διάνυσμά του και υπολογίζεται ως φυσιολογική τιμή προσθέτοντας στο πίνακά του με πιθανότητα N/M_2. Επίσης και το διάνυσμα του

κόμβου 5 θεωρείται από το κόμβο δύο ως φυσιολογική μέτρηση και προστίθεται και αυτό στο πίνακά του με πιθανότητα N/M_2

5.6.2 Ανάλυση λειτουργίας πινάκων κατακερματισμού και συναρτήσεών του της τεχνικής LSH

Η τεχνική μας υποστηρίζει τρεις συναρτήσεις, τη συνάρτηση *insert()* για την εισαγωγή ενός καινούριου διανύσματος μετρήσεων στον κόμβο, τη συνάρτηση *delete()* για τη διαγραφή ενός διανύσματος μετρήσεων από το κόμβο και τη συνάρτηση *checkIfOutlier()* για τον έλεγχο ενός διανύσματος εάν είναι ακραίο σ'ένα κόμβο. Το σύστημα μας από πίνακες κατακερματισμού αποτελείται από *L* πίνακες κατακερματισμού όσες είναι και οι οικογένειες συναρτήσεων κατακερματισμού και από έναν αριθμό κουβάδων στον κάθε πίνακα που ορίζεται στην αρχή του προγράμματός μας. Η κάθε οικογένεια συναρτήσεων αποτελείται από *k* συναρτήσεις όπως έχει αναλυθεί στο προηγούμενο κεφάλαιο. Ένας κουβάς του πίνακα είναι μία λίστα από διανύσματα που εκχωρούνται. Στην πραγματικότητα αποθηκεύεται σ'έναν κόμβο της λίστας ο αριθμός ταυτότητας του διανύσματος και ο αριθμός ταυτότητας της συνάρτησης κατακερματισμού με την οποία εκχωρήθηκε σ'αυτό το κουβά.



Σχήμα 5.5 Στο α) είναι η δομή ενός πινάκα κατακερματισμού του LSH σχήματος μας και στο β) ο κόμβος ενός κουβά του πίνακα

Αναλυτικότερα οι συναρτήσεις που υποστηρίζει η τεχνική μας είναι:

- *insert()* : Δέχεται ως όρισμα το μοναδικό αριθμό ταυτότητας του διανύσματος που είναι και ο αριθμός θέσης του στο πίνακα διανυσμάτων του κόμβου που θα εγγραφτεί. Η διαδικασία είναι ως εξής για κάθε συνάρτηση κατακερματισμού της οικογένειας

$$h_{a,b}(v) = \left\lfloor \frac{a \bullet v + b}{r} \right\rfloor$$

  βρίσκεται ο κουβάς που θα αποθηκευτεί το διάνυσμα αποθηκεύοντας τον αριθμό ταυτότητας του διανύσματος και τον αριθμό της συνάρτησης που χρησιμοποιήθηκε στη συνάρτηση κατακερματισμού.
- *delete()* : Δέχεται ως όρισμα το μοναδικό αριθμό ταυτότητας του διανύσματος. Από τον αριθμό αυτο βρίσκουμε στο πίνακα του κόμβου το διάνυσμα με τις μετρήσεις και με βάση αυτές κάνουμε την ίδια διαδικασία με τη συνάρτηση *insert()* για να βρεθεί σε ποιους κουβάδες είχε εισαχθεί. Στους κουβάδες αυτούς

γίνεται διαγραφή κόμβου στη λίστα του κουβά που είχε δημιουργηθεί για το διάνυσμα αυτό.

- *checkIfOutlier()* : Δέχεται ως όρισμα ένα διάνυσμα μετρήσεων και επιστρέφει *1* εάν το διάνυσμα είναι ακραίο για το συγκεκριμένο κόμβο και *0* εάν το διάνυσμα δεν είναι ακραίο. Κάνουμε την ίδια διαδικασία σαν να εισάγεται το διάνυσμα αυτό και βρίσκουμε τους κουβάδες που θα εισαγότανε. Για να μην θεωρηθεί ως ακραίο το διάνυσμα θα πρέπει οι κουβάδες που θα εισαγότανε να έχουν τουλάχιστον το κατώφλι υποστηρικτών(οι υποστηρικτές είναι μοναδικοί) που έχει οριστεί ως κατώφλι στην αρχή του προγράμματος.

5.7 Λειτουργία άπληστης μεθόδου

Ο κάθε κόμβος όπως και στη μέθοδο μας θα έχει ένα πίνακα για να αποθηκεύει διανύσματα από μετρήσεις του ή μετρήσεις παιδιών του. Με τον ίδιο τρόπο ψάχνουμε για ακραίες μετρήσεις στους κόμβους(από τα φύλλα προς τα πάνω στο δέντρο ως την ρίζα) μόνο που στην προκειμένη περίπτωση η αναζήτηση που θα κάνουμε δεν θα βασίζεται στο LSH σχήμα αλλά θα γίνεται σειριακή αναζήτηση στο πίνακα των διανυσμάτων.

5.8 Λειτουργία κεντρικοποιημένης μεθόδου

Στη κεντρικοποιημένη μέθοδο υπάρχει ένας κεντρικός πίνακας αποθήκευσης διανυσμάτων μεγέθους όσος ήτανε στο κάθε κόμβο ο πίνακας επί τον αριθμό των κόμβων του δικτύου. Ο πίνακας αυτός βρίσκεται στο σταθμός βάσης(ή τη ρίζα του δέντρου) στον οποίο θα γίνεται η κεντρική επεξεργασία για την εύρεση των ακραίων τιμών όλων των κόμβων του δικτύου χωρίς το LSH. Η διαδικασία για κάθε εποχή είναι η εξής:

- Όλοι οι κόμβοι στέλνουν τις μετρήσεις μίας εποχής στο σταθμό βάσης(ή ρίζα του δέντρου)
- Τα καινούρια διανύσματα ανεξάρτητα εάν είναι ή όχι ακραία αντικαθιστούν παλιά διανύσματα με δύο τρόπους

  α΄ τρόπος: Σε μία τυχαία θέση του πίνακα

  β’ τρόπος: Στο παλαιότερο διάνυσμα

- Γίνεται έλεγχος των νέων διανυσμάτων στο κεντρικό πίνακα εάν είναι ακραίο με την ελάχιστη υποστήριξη που έχουμε από την παράμετρο *minsupport*
- Σε κάθε εποχή τα τρία αυτά βήματα επαναλαμβάνονται

Η κεντρικοποιημένη μέθοδος δίνει τα αποτελέσματα με 100% αξιοπιστία και είναι ένα μέτρο σύγκρισης με τις άλλες δύο μεθόδους για την αξιοπιστία των αποτελεσμάτων τους.

5.9 Μέτρηση bytes που στέλνονται στο δίκτυο αισθητήρων

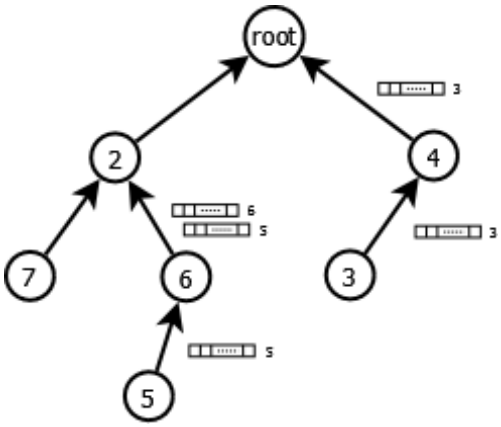
Η αποστολή δεδομένων από ένα κόμβο σ'ένα άλλον πρέπει να είναι όσο το δυνατόν μικρότερη, ώστε να πετυχαίνουμε μικρότερη κατανάλωση ενέργειας που συνεπάγεται και μακροζωία χωρίς όμως να χάνεται η αξιοπιστία των αποτελεσμάτων του δικτύου. Έτσι ο υπολογισμός των bytes είναι σημαντικός παράγοντας για την σύγκριση μεθόδων.

Θα ορίσουμε την αποστολή bytes από έναν κόμβο παιδί στο κόμβο πατέρα του. Οι πληροφορίες που στέλνονται είναι τα διανύσματα που θεώρησε ο κόμβος ως ακραίες τιμές και τους αριθμούς ταυτότητας των κόμβων που έχουν θεωρηθεί ακραίοι. Οι πραγματικοί αριθμοί και οι ακέραιοι αριθμοί που στέλνονται έχουν αντίστοιχα μέγεθος 8 bytes και 4 bytes. Το διάνυσμα μας περιέχει πραγματικούς αριθμούς και ο αριθμός ταυτότητας του κόμβου είναι ακέραιος αριθμός. Με βάση λοιπόν τα παραπάνω που ορίστηκαν ο τύπος για τα bytes που στέλνει ένας κόμβος παιδί στο κόμβο πατέρα του είναι

$$[n \times d \times (8 \text{ bytes})] + [\text{n} \times (4 \text{ bytes})]$$

όπου n είναι ο αριθμός των ακραίων διανυσμάτων που έχει θεωρήσει ο κόμβος , d είναι ο αριθμός των πραγματικών αριθμών που περιέχει το κάθε διάνυσμα και τέλος το μέγεθος σε bytes του κάθε αριθμού και του αριθμού ταυτότητας.

Παράδειγμα 5.3 Έχουμε ένα δίκτυο επτά αισθητήρων. Ως ακραίες μετρήσεις έχουν οι κόμβοι 5 , 6 και 3 στο δίκτυο που δίνεται παρακάτω και το διάνυσμα του κάθε κόμβου περιέχει οκτώ μετρήσεις. Θα παρουσιαστεί το σύνολο bytes που στάλθηκαν σε αυτή την εποχή.



Σχήμα 5.6 Σχηματικό διάγραμμα λειτουργίας της μεθόδου μας

Στο σχήμα μας έχουμε πέντε διανύσματα που στάλθηκαν σ΄αυτήν την εποχή και το κάθε διάνυσμα περιέχει οκτώ μετρήσεις. Με βάση τον τύπο που παρουσιάστηκε παραπάνω θα έχουμε

$$[5\times 8\times (8\text{ bytes})] + [5\times (4\text{ bytes})] = 340\text{ bytes}$$

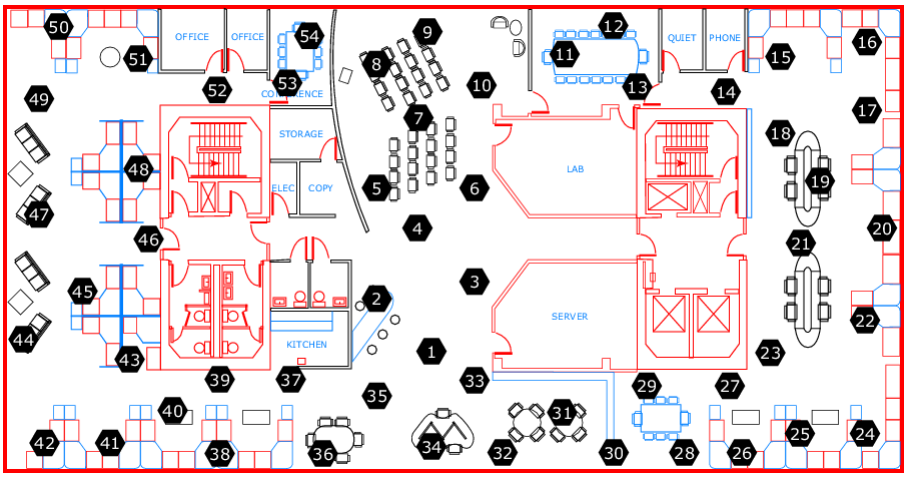
ΚΕΦΑΛΑΙΟ 6

ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ

Για ανακεφαλαίωση αυτά που υλοποιήθηκαν στην εργασία αυτή είναι η δικιά μας μέθοδος, η άπληστη μέθοδος και η κεντρικοποιημένη μέθοδος η οποία χρησιμοποιείται για τον έλεγχο της αξιοπιστίας των δύο προηγούμενων μεθόδων. Για να αξιολογηθούν οι μέθοδοι θα πρέπει να υπάρχει μια κοινή βάση δεδομένων και να τραβιούνται τα δεδομένα με την ίδια διαδικασία. Για να γίνει ρεαλιστική αξιολόγηση των μεθόδων με ρεαλιστικές μετρήσεις κρίθηκε απαραίτητο να μην παραχθούν τυχαία δεδομένα παρά να χρησιμοποιηθεί μια πραγματική βάση μετρήσεων από αισθητήρες ενός μεσαίου μεγέθους ΑΔΑ.

6.1 Περιγραφή των δεδομένων που χρησιμοποιούνται

Αποφασίστηκε να χρησιμοποιηθούν τα δεδομένα από το ΑΔΑ που υπάρχει στο Intel Berkeley Research Lab[12]. Στο ΑΔΑ αυτό υπάρχουν 54 ασύρματοι αισθητήρες που είναι κατανεμημένοι στο εργαστήριο όπως φαίνεται στην παρακάτω εικόνα



Εικόνα 6.1 Κάτοψη του Intel Research Berkeley με το ΑΔΑ

Στην ιστοσελίδα που βρίσκεται στο διαδίκτυο[12] δίνονται οι μετρήσεις όλων των αισθητήρων του ΑΔΑ που περιέχουν τις εξής πληροφορίες μαζί με τις μονάδες μετρήσεις τους:

- Υγρασία (0-100%)
- Φωτεινότητα (0-100.000 LUX)
- Θερμοκρασία (Κελσίους)
- Τάση(ηλεκτρική) (2-3 Volt)

οι μετρήσεις αυτές έχουν παρθεί το χρονικό διάστημα από 28 Φεβρουαρίου του 2004 έως τις 5 Απριλίου του ίδιου χρόνου παίρνοντας μετρήσεις ανά 31 δευτερόλεπτα. Επίσης από το εργαστήριο δίνεται ένα αρχείο με τις συντεταγμένες *x* και *y* θέτοντας ως αρχή των αξόνων το σημείο πάνω δεξιά της εικόνας 6.1. Παρακάτω παρουσιάζεται στη βάση δεδομένων την μορφή που έχουν τα δεδομένα

Ημερομηνία | *ώρα* | *εποχή*| *αριθμός κόμβου* | *θερμοκρασία* | *υγρασία* | *φωτεινότητα* | *τάση*

Για λόγους απλότητας, μείωσης του υπολογιστικού χρόνου και ποιο γρήγορης σύγκρισης μεθόδων αποφασίστηκε να επιλεχθούν οι πρώτοι 32 αισθητήρες και να απομονωθεί από όλα τα δεδομένα η θερμοκρασία μόνο η οποία μπορεί να παρουσιάσει ενδιαφέροντα δεδομένα για παρατήρηση. Επίσης κάθε αισθητήρας θεωρείται και ένας κόμβος του δέντρου. Άρα τα δεδομένα χωρίστηκαν κατά αριθμό κόμβου(φτιάχνοντας *txt* αρχεία με όνομα τον αριθμό του κόμβου) βάζοντας ημερομηνιακά τα δεδομένα αντίστοιχα στο κόμβο που αντιστοιχούν. Οι μέθοδοι αντλούν τις μετρήσεις από τα αρχεία σειριακά.

6.2 Μέτρα σύγκρισης

Για να συγκριθούν και για να αξιολογηθούν οι μέθοδοι συγκρίνουμε την ενέργεια που χρειάζονται για την επικοινωνία μεταξύ τους, την αξιοπιστία που έχουν, το χώρο μνήμης που απαιτούν και το χρόνο εκτέλεσης τους. Αυτές είναι οι τέσσερις βασικές παράμετροι αξιολόγησης και συγκρισή τους.

Η ενέργεια το μεγαλύτερο κομμάτι όπως έχει προαναφερθεί στο παραπάνω κεφάλαιο, βρίσκεται κατά ένα μεγάλο μέρος στην επικοινωνία που γίνεται μεταξύ των κόμβων αισθητήρων και μετριέται στον αριθμό των bytes που στέλνονται. Αυτό μπορεί να υποτεθεί γιατί τα ποσά είναι ανάλογα(π.χ. λιγότερη αποστολή bytes σημαίνει και μικρότερη κατανάλωση ενέργειας).

Για την μέτρηση αξιοπιστίας των δύο μεθόδων χρησιμοποιούνατι δύο παραμέτροι η «ακρίβεια»(*precision*) και η «ανάκληση»(*recall*) σε σχέση με την κεντρικοποιημένη μέθοδο που θεωρείται ότι έχει τα πραγματικά αποτελέσματα. Οι σχέσεις που δίνονται για την ακρίβεια και ανάκληση είναι:

$$\text{Precision} = \frac{\text{Πραγματικά ακραία διανύσματα της μεθόδου}}{\text{Συνολικός αριθμός ακραίων διανυσμάτων της μεθόδου}}$$

$$\text{Recall}=\frac{\text{Πραγματικά ακραία διανύσματα της μεθόδου}}{\text{Συνολικός αριθμός πραγματικών ακραίων διανυσμάτων}}$$

Ουσιαστικά όταν υπάρχει λανθασμένη απόφαση για το εάν ένα διάνυσμα ήταν ακραίο υπάρχουν δύο περιπτώσεις. Η μία περίπτωση λέγεται «ψευδές θετικό αποτέλεσμα» δηλ. το αποτέλεσμα ήτανε ακραίο διάνυσμα ενώ δεν ήταν στην πραγματικότητα και η άλλη περίπτωση «ψευδές αρνητικό αποτέλεσμα» δηλ. το αποτέλεσμα ήτανε φυσιολογικό διάνυσμα ενώ ήταν ακραίο στην πραγματικότητα. Αυτές οι δύο περιπτώσεις λαθών μπορούν να αναπαρασταθούν από τις δύο παραμέτρους ανάκληση και ακρίβεια. Η παράμετρος ακρίβεια είναι το μέγιστο όταν δεν υπάρχουν «ψευδές θετικό αποτέλεσμα» και η ανάκληση είναι μέγιστο όταν δεν υπάρχουν «ψευδές αρνητικό αποτέλεσμα». Επομένως για υψηλή ακρίβεια τα αποτελέσματα που χαρακτηρίζονται ως ακραία είναι και σωστά αλλά δεν γνωρίζεται κατά πόσο χάθηκαν αποτελέσματα που ήταν ακραία διανύσματα ενώ αντίστροφα για υψηλή ανάκληση βρίσκονται τα περισσότερα ακραία διανύσματα αλλά δεν γνωρίζεται κατά πόσο έχει επιστρέψει αποτελέσματα ως ακραία ενώ δεν ήτανε. Άρα σε συνδυασμό των δύο παραμέτρων μπορεί να αξιολογηθεί η αξιοπιστία των μεθόδων και σε τι υστερούν.

Ο χώρος μνήμης που καταλαμβάνεται στη κάθε μέθοδο είναι οι πίνακες κατακερματισμού και ο πίνακας που αποθηκεύει τα δεδομένα(μετρήσεις).

Ο χρόνος εκτέλεσης μίας μεθόδου που ορίζεται στην εργασία μας για λόγους απλότητας είναι από την στιγμή που θα εισάγει το πρώτο δεδομένο στο κόμβο μέχρι το τελείωμα όλων των εποχών που έχει οριστεί για το πείραμα. Στην πραγματικότητα όμως ο χρόνος για τις τεχνικές του LSH σχήματος και της άπληστης μεθόδου είναι διαφορετικός. Στο LSH σχήμα και στην άπληστη μέθοδο υλοποιούνται σένα δέντρο αισθητήρων και η επεξεργασία των μετρήσεων γίνεται σε κάθε κόμβο ενώ στην κεντρικοποιημένη μέθοδο όλες οι επεξεργασίες γίνονται στο κόμβο ρίζα του δέντρου. Ως αποτέλεσμα έχει στις δύο πρώτες τεχνικές η επεξεργασία να γίνεται παράλληλα στους κόμβους σε κάθε επίπεδο άρα ως πραγματικός χρόνος ορίζεται το άθροισμα του μέγιστου χρόνου κόμβου που γίνεται σε κάθε επίπεδο. Έτσι ουσιαστικά οι χρόνοι θα ήταν μικρότεροι και από τους χρόνους που παρουσιάζονται παρακάτω στα πειράματα.

6.3 Πειράματα

Για να γίνουν τα πειράματα τέθηκαν κάποιες αρχικές τιμές στις παραμέτρους:

delta 0.90
R 1.5
minsupport 10
size_batch 5
p 1
numOfBucketsPerHashTable 800

numOfVectorsInDataSet 800
probabilitySentParentNode 0.1
dynamicDataSet 1
factorDynamicDataSet 0.1
broadcastDist 17
read_epoch 3
size_int 4
size_float 8
withOrNotSlidingWindow 0
slidingWindow_w 1200
run_epochs 1200

Με τις παραπάνω τιμές στις παραμέτρους τα αποτελέσματα είναι:

Preciosion_OWN = 0,84
Recall_OWN = 0,41
time_OWN = 45

Precision_GREEDY = 0.59
Recall_GREEDY = 0.29
time_ GREEDY = 20

και το πραγματικό ποσοστό ακραίων τιμών στο σύνολο των δεδομένων ήτανε 6% των τιμών.

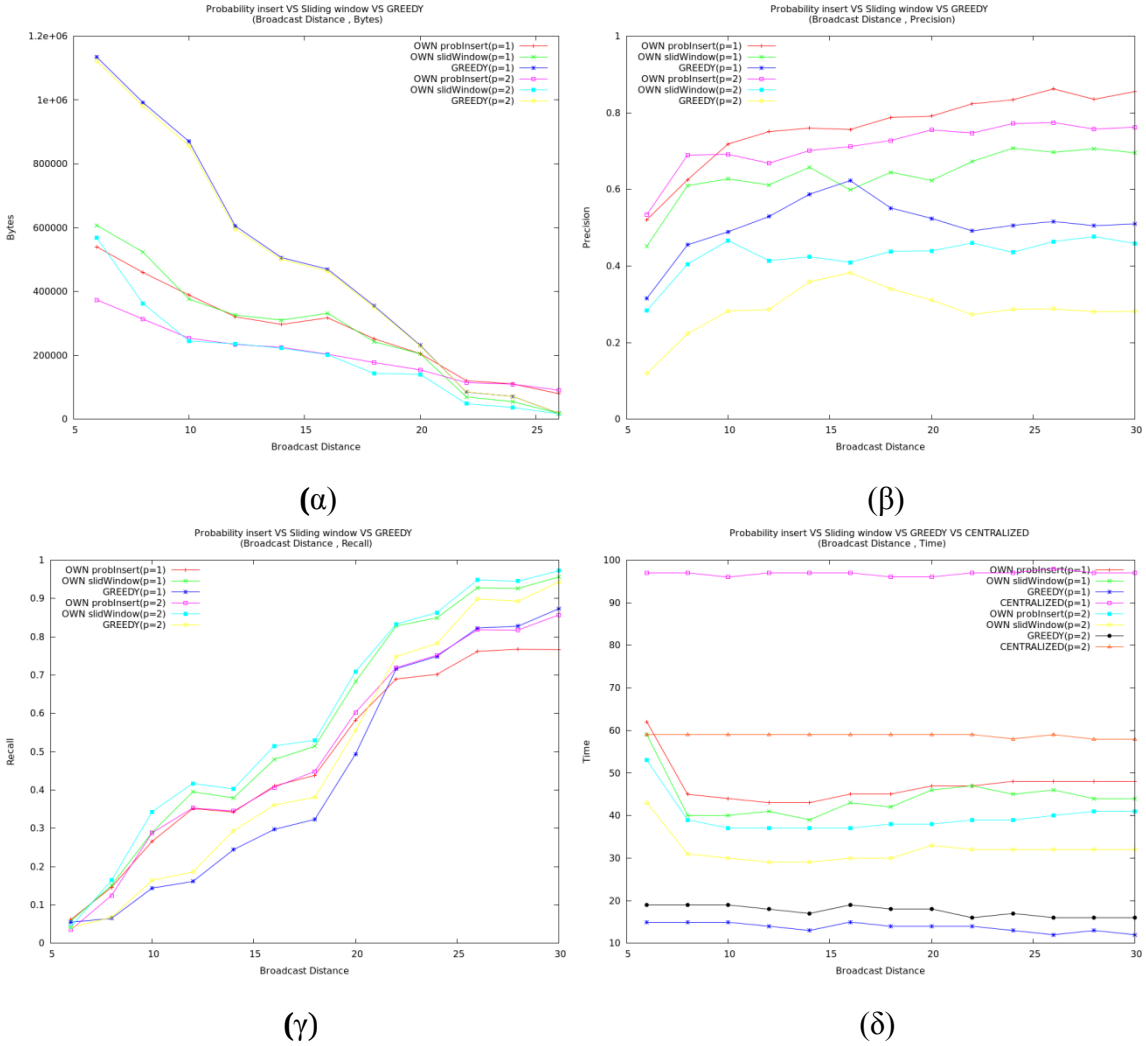
Για το delta = 0.9 η συνάρτηση εύρεσης βέλτιστων παραμέτρων k και L έβγαλε ως βέλτιστα L=15 και k=14.

Τα πειράματα θα γίνουν ως εξής : κρατώντας σταθερές τις παραπάνω τιμές των παραμέτρων θα μεταβάλλεται κάθε φορά μία παράμετρος για να δειχθεί η συμπεριφορά των μεθόδων ως προς την παράμετρο και για την σύγκριση μεταξύ τους. Ουσιαστικά στα πειράματα συγκρίνονται τέσσερις μεθόδοι. Οι δύο είναι οι δικές μας η «πιθανότητα εισαγωγής» και το «ολισθημένο παράθυρο» οι οποίες βασίζονται στο LSH σχήμα με διαφορετικό όμως τρόπος εισαγωγής των δεδομένων στο σύνολο δεδομένων του αισθητήρα. Οι άλλες δύο είναι η άπληστη μέθοδος και η κεντρικοποιημένη όπου η τελευταία χρησιμοποιείται για τη σύγκριση των άλλων μεθόδων(παρουσιάζεται όμως στα αποτελέσματα των πειραμάτων ως προς την αποστολή bytes και το χρόνο εκτέλεση της).

6.3.1 Πείραμα 1ο

Στο πρώτο πείραμα μεταβάλλεται η απόσταση μεταξύ δύο κόμβων-αισθητήρων για να θεωρούνται γείτονες(*broadcastDist*). Αυτό σημαίνει ότι όσο αυξάνεται η τιμή της

παραμέτρου τόσο μειώνεται επακόλουθα το βάθος του δέντρου αισθητήρων και έτσι δείχνεται πως επηρεάζει το βάθος του δέντρου τις μεθόδους.



Εικόνα 6.2 Συναρτήσει του broadcast Distance με την αποστολή bytes, την ακρίβεια, την ανάκληση και το χρόνος εκτέλεσης

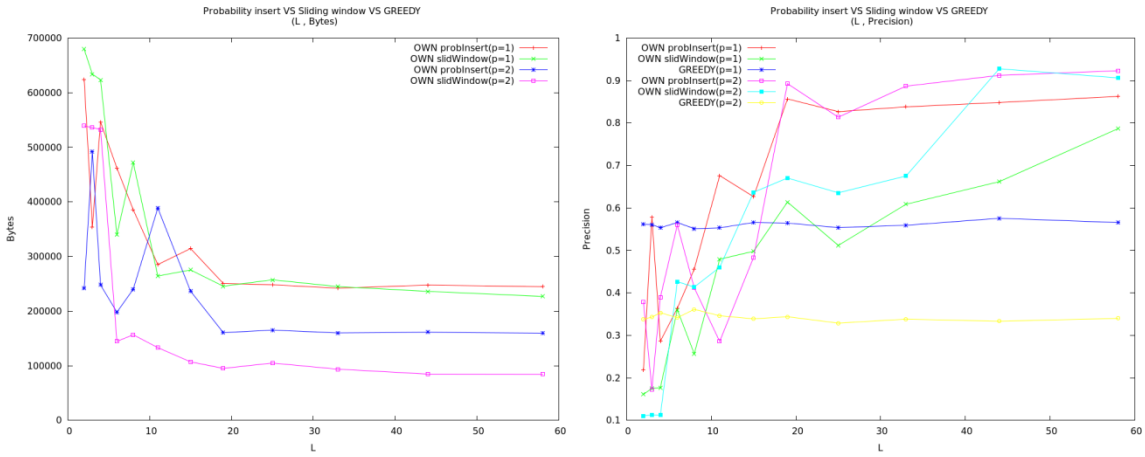
Στην εικόνα 6.2 (α) η αποστολή bytes της κεντρικοποιημένης μέθοδου δεν παρουσιάζεται γιατί τα αποτελέσματά της είναι μία με δύο τάξης μεγαλύτερη από τις υπόλοιπες μεθόδους(Αυτό γίνεται σε όλα τα πειράματα). Οι υπόλοιπες μέθοδοι παρατηρείται γενικά ότι είναι κοντά στην αποστολή bytes. Αρχικά η δύο τρόποι της μεθόδους μας για μεγαλύτερο βάθος δέντρου στέλνουν λιγότερα bytes και από την άπληστη μέθοδο. Στην πραγματικότητα οι δύο δικές μας μέθοδοι «πιθανότητα εισαγωγής» και «ολισθημένο παράθυρο» στέλνουν πολύ μικρότερο όγκο δεδομένων γιατί χρησιμοποιούν το LSH σχήμα άλλα κρίθηκε προτιμότερο να χρησιμοποιούμε αποστολή

ένα ποσοστό δεδομένων προς τα πάνω στο δέντρο για να κρατιέται. Έχει ως αποτέλεσμα ανάλογα με το ποσοστό που χρησιμοποιείται να αυξάνεται ο όγκος των δεδομένων που στέλνονται μεταξύ των κόμβων. Στα πειράματα μας κρίθηκε μία καλή τιμή *probabilitySentParentNode 0.1*. Στην εικόνα 6.2 (β),(γ) παρατηρείται μια μικρή αύξηση της ακρίβειας όσο το βάθος του δέντρου μικραίνει αλλά με καλύτερα αποτελέσματα σαφώς η δικιά μας μέθοδος «πιθανότητα εισαγωγής» και μία γραμμική αύξηση της ανάκλησης σε όλους τις μεθόδους όσο μειώνεται το βάθος του δέντρου. Ένας άλλος λόγος που βρίσκονται περισσότερα ακραία διανύσματα(φαίνεται από την αύξηση της ανάκλησης) είναι ότι η μέθοδος με την αύξηση της απόστασης κεντρικοποιείται δηλ. αποκτάει λιγότερους πατέρες με περισσότερα παιδιά. Τέλος στην εικόνα 6.2 (δ) φαίνεται ότι ο χρόνος εκτέλεσης δεν εξαρτάται από το βάθος του δέντρου και ότι σε απόδοση τη χειρότερη έχει η κεντρικοποιημένη μετά η δική μας μέθοδος και την καλύτερη η άπληστη μέθοδος.

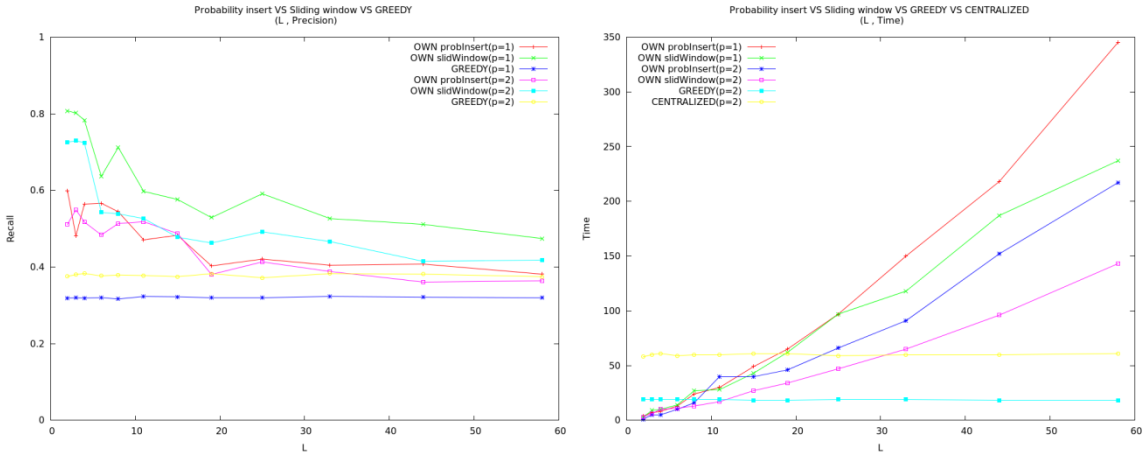
Το σύμπερασμα ότι όσο αυξάνεται η απόσταση για να θεωρούνται δύο κόμβοι γειτονικοί καλυτερεύουν τα αποτελέσματα και αυτό τείνει όμως να γίνεται κεντρικοποιημένη η μέθοδος μας με ένα κόμβο πατέρα και οι υπόλοιποι κόμβοι παιδιά. Το πρόβλημα στην πραγματικότητα που δημιουργείται είναι ότι αυξάνοντας την απόσταση πολύ αυξάνεται και η ενέργεια που χρειάζονται δύο κόμβοι να επικοινωνήσουν, πράγμα το οποίο είναι πολύ σημαντικο γιατί όπως έχει αναφερθεί στην αρχή της εργασίας οι αισθητήρες έχουν περιορισμένη ενέργεια. Έτσι διαλέγεται η απόσταση με κριτήριο και την ενέργεια που θα χρειάζεται ο κάθε κόμβος για την επικοινωνία.

6.3.2 Πείραμα 2ο

Το δεύτερο πείραμα μεταβάλει τον αριθμό k που δείχνει τον αριθμό των συναρτήσεων που έχει μία οικογένεια του LSH σχήματος. Από το k έχει αναφερθεί στο προηγούμενο κεφάλαιο για την εύρεση των βέλτιστων k και L(ο αριθμός των οικογενειών που έχει το LSH σχήμα). Στις εικόνες στον χ-άξονα παρουσιάζεται μόνο ο αριθμός των οικογενειών-πίνακες κατακερματισμου L.



(α) (β)



(γ) (δ)

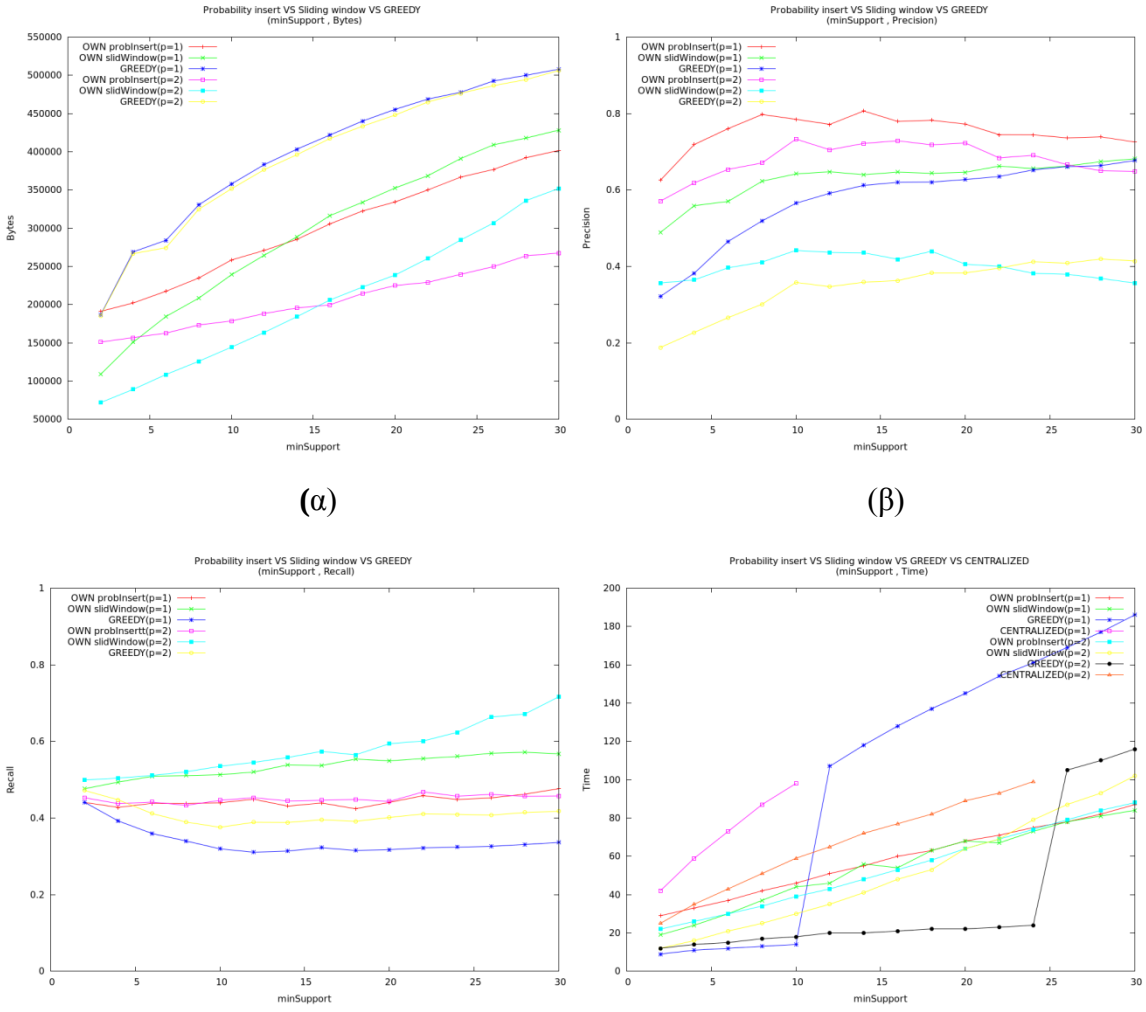
Εικόνα 6.3 Συναρτήσει του L(αριθμός οικογενειών του LSH σχήματος-αριθμός πίνακας κατακερματισμού) με την αποστολή bytes, την ακρίβεια, την ανάκληση και το χρόνο εκτέλεσης

Καταρχήν νόημα για τη μεταβολή του L και το k έχει μόνο οι δύο δικές μας μεθόδοι «πιθανότητα εισαγωγής» και «ολισθημένο παράθυρο» που χρησιμοποιούν το LSH σχήμα. Στις άλλες μεθόδους τα αποτελέσματα είναι σταθερα γιατί δεν χρησιμοποιούν το LSH και δεν εξαρτώνται από το L και k. Παρατηρείται σε όλες τις εικόνες του 6.3 ότι το L κάτω από 15 δεν εξάγει σταθερά αποτελέσματα οπότε και δεν σχολιάζονται. Στην εικόνα 6.3 (α) παρατηρείται ότι στέλνονται λιγότερα bytes στις μεθόδους μας για L>15. Στην εικόνα 6.3 (β),(γ) ειδικά η μέθοδός μας «πιθανότητα εισαγωγής» έχει αρκετά καλύτερα αποτελέσματα από την άπληστη μέθοδος. Τέλος στην εικόνα 6.3 (δ) φαίνεται ότι ο χρόνος εκτέλεσης στις μεθόδους μας χειροτερεύει γραμμικά με την αύξηση του L.

Το συμπέρασμα είναι μέχρι εκεί που επιτρέπει ο χρόνος εκτέλεσης περίπου στα ίδια επίπεδα με της άπληστης μεθόδου για L κοντά στο 18 υπάρχουν καλύτερα αποτελέσματα στην ακρίβεια και στην ανάκληση από την άπληστη μέθοδος.

6.3.3 Πείραμα 3ο

Το τρίτο πείραμα μεταβάλλει το κατώφλι των υποστηρικτών για να θεωρείται ένα διάνυσμα φυσιολογικό. Όσο αυξάνεται το κατώφλι τόσο πιο εύκολα ένα διάνυσμα θα χαρακτηρίζεται ακραίο και θα υπάρχει μεγάλο ποσοστό των διανυσμάτων που θεωρούνται ακραίες τιμές για το δίκτυο μας. Σένα πραγματικό δίκτυο όμως δε γίνεται να υπάρχουν πολλά ακραία διανύσματα δηλαδή να έχουμε μεγάλο αριθμό κατωφλίου υποστηρικτών. Έτσι δίνεται περισσότερο βάρος για μικρό αριθμό υποστηρικτών.

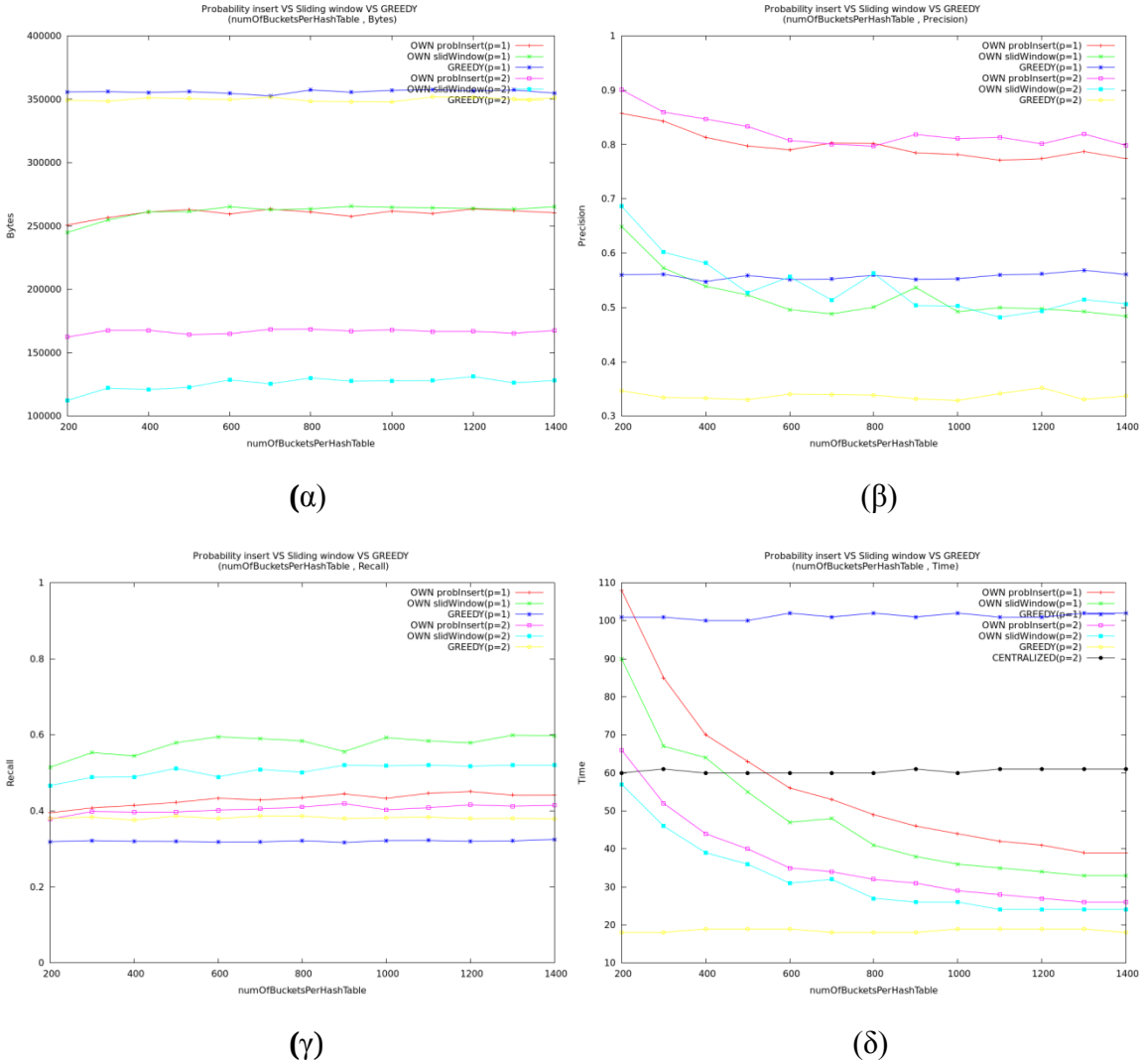


Εικόνα 6.4 Συναρτήσει του κατωφλίου υποστηρικτών με την αποστολή bytes, την ακρίβεια, την ανάκληση και το χρόνος εκτέλεσης

Παρατηρείται στην εικόνα 6.4(α) ότι όσο μεγαλώνει το κατώφλι των υποστηρικτών η αποστολή των bytes αυξάνεται και αυτό είναι λογικό γιατί έχοντας περισσότερες ακραίες τιμές χρειάζεται να στέλνονται στο δέντρο στο παραπάνω επίπεδο κάθε φορά. Όμως οι μεθόδοι του LSH σχήματος παρατηρείται ότι είναι πιο ανθεκτικές στην αύξηση ακραίων τιμών στο δίκτυο των αισθητήρων. Στην εικόνα 6.4(β) που δείχνει την ακρίβεια βλέπουμε ότι η «πιθανότητα εισαγωγής» έχει καλύτερη απόδοση από τις άλλες μεθόδους ενώ στην εικόνα 6.4(γ) που δείχνει την ανάκληση βλέπουμε ότι συμβαίνει το ίδιο για το «ολισθημένο παράθυρο». Στην εικόνα 6.4(δ) φαίνεται ότι ο χρόνος όπου αυξάνεται γραμμικά μαζί με την αύξηση του κατωφλίου.

6.3.4 Πείραμα 4ο

Στο τέταρτο πείραμα μεταβάλλεται ο αριθμός των κουβάδων που έχει ο κάθε πίνακας κατακερματισμού του LSH σχήματος. Αυξάνοντας τον αριθμό των κουβάδων αυξάνεται και η μνήμη ουσιαστικά που χρειάζεται ο κάθε αισθητήρας αλλά μειώνεται και ο αριθμός των κόμβων-λίστας που υπάρχει σε κάθε κουβά. Αυτό συνεπάγεται με μείωση του εκτελεστικού χρόνου.



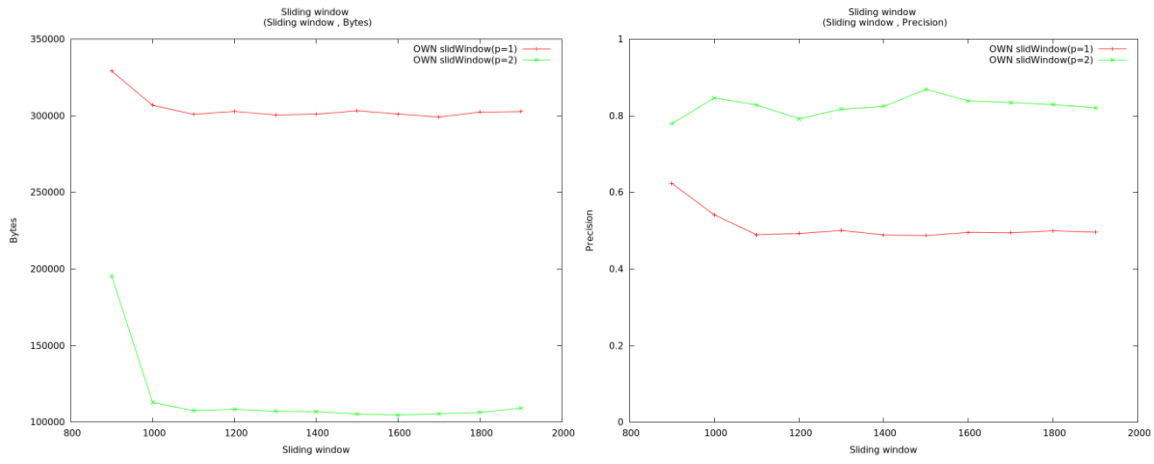
Εικόνα 6.5 Συναρτήσει του αριθμού των κουβάδων ανά πίνακα κατακερματισμού με την αποστολή bytes , την ακρίβεια την ανάκληση και το χρόνο εκτέλεσης

Όπως και στο 2ο πείραμα που μεταβάλλεται το L και k αυτά που συγκρίνονται είναι οι δικές μας μέθοδοι που βασίζονται στο LSH σχήμα ενώ οι υπόλοιπες μέθοδοι παραμένουν σταθερές. Στην εικόνα 6.5(α) όπως και είναι λογικό δεν παρατηρείται κάποια μεταβολή στην αποστολή των bytes. Παρατηρείται στις εικόνες 6.5(β)(γ) ότι αυξάνοντας τον αριθμό των κουβάδων ανα πίνακα κατακερματισμού η ποιότητα των μεθόδων ως προς την ακρίβεια και την ανάκληση μεταβάλλονται λίγο(σχεδόν σταθερές). Αφού δεν μεταβάλλονται ουσιαστικά η αποστολή bytes η ακρίβεια και η ανάκληση σημαίνει ότι ανιχνεύονται ακραία διανύσματα τα ίδια και δεν επηρεάζεται από την αύξηση των κουβάδων. Ως προς τον χρόνο εκτέλεσης στην εικόνα 6.5(δ) παρατηρείται η σημαντική μείωση του χρόνου αυξάνοντας των αριθμό των κουβάδων.

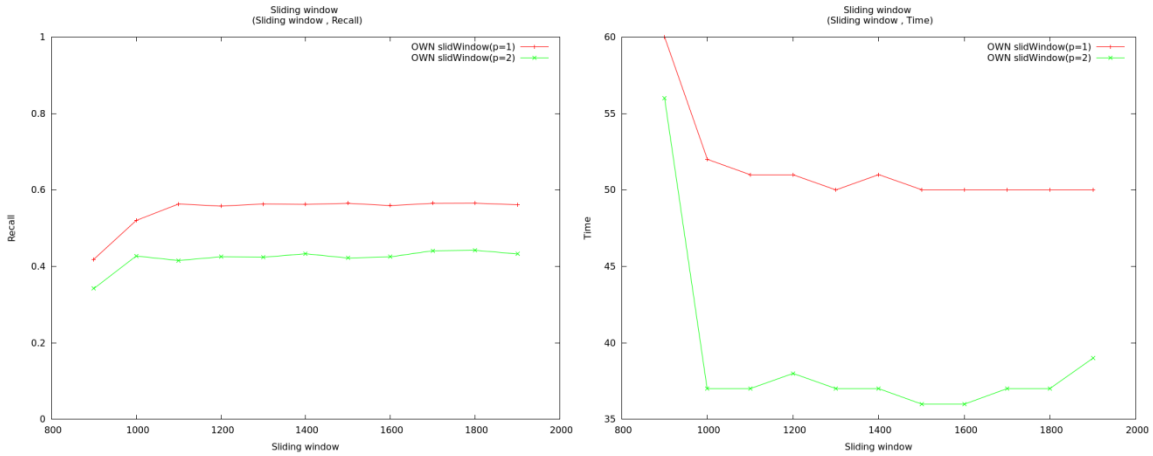
Για το εύρος που έγινε το πείραμα και έχοντας σταθερή την απόδοση των μεθόδων βάζουμε μεγάλο αριθμό κουβάδων ανά πίνακα κατακερματισμού στα πειράματά μας για να έχουμε όσο το δυνατόν χαμηλό χρόνο εκτέλεσης. Το πρόβλημα που μπορεί να υπάρξει είναι το μέγεθος μνήμης ενός αισθητήρα. Λόγω όμως ότι η μνήμη στην σήμερινη εποχή δεν κοστίζει και μπορούμε να έχουμε σ'ένα αισθητήρα αρκετά μεγάλη μνήμη(γιαυτό που χρειάζεται) δεν επηρεάζεται τόσο η απόφαση του αριθμού των κουβάδων στο LSH σχήμα μας.

6.3.5 Πείραμα 5ο

Στο 5ο πείραμα και τελευταίο παρατηρείται η απόδοση της τεχνικής ολισθημένου παράθυρου ανάλογα με το μέγεθος του παραθύρου που χρησιμοποιείται. Όσο αυξάνεται το παράθυρο μειώνεται η πιθανότητα εισαγωγής μίας τιμής στο πίνακα δεδομένων του αισθητήρα.



(α) (β)



(γ) (δ)

Εικόνα 6.6 Συναρτήσει του μεγέθους του παραθύρου με την αποστολή bytes , την ακρίβεια την ανάκληση και το χρόνο εκτέλεσης

Παρατηρείται ότι το ολισθημένο παράθυρο είναι ανθεκτικό και δεν επηρεάζεται με την αύξηση του παραθύρου. Η απόδοση όμως που έχει στην ακρίβεια και στην ανάκληση είναι αρκετά χαμηλή γι'αυτό η μέθοδος με «πιθανότητα εισαγωγής» είναι καλύτερη.

6.4 Συμπεράσματα

Συμπεράσματα για την κάθε παράμετρο πως μπορεί να επηρεάσει τις μεθόδους γράφτηκαν παραπάνω στο κάθε πείραμα ξεχωριστα. Μέσω όμως διάφορων τεχνικών προσπαθήσαμε να βελτιώσουμε την απόδοση των τεχνικών αλλά δεν υπήρχαν επαρκή αποτελέσματα. Καταρχήν υλοποιήθηκαν δύο μεθόδοι πάνω στο σχήμα μας η «πιθανότητα εισαγωγής» και το «ολισθημένο παράθυρο» στις οποίες εφαρμόστηκαν διάφορες μικροτεχνικές για την καλυτέρευση των αποτελεσμάτων. Είναι δύο τεχνικές που λειτουργούν διαφορετικά στο τρόπο που κρατάνε τα αποτελέσματα. Κάποιες από αυτές παρουσιάζονται παρακάτω:

- scaling(κλιμάκωση) του κατωφλίου υποστηρικτών ανάλογα με τις φυσιολογικές τιμές που έχουν περάσει από το κόμβο και το μέγεθος του πίνακα δεδομένων που έχει ο πίνακας.
- μέσω μίας πιθάνοτητας εισαγωγής διανυσμάτων στο παραπάνω επίπεδο στο δέντρο επιτεύχθηκε η καλύτερη ενημέρωση των κόμβων στο δέντρο. Επίσης λόγω του τελευταίου δημιουργήθηκε και το δυναμικό μέγεθος του πίνακα δεδομένων στο δέντρο(το μέγεθος του πίνακα εξαρτάται από πόσους κόμβους παιδιά έχει ένας κόμβος).

- χρησιμοποιήθηκαν για την εύρεση υποστηρικτών σένα κόμβο και η λίστα με τα ακραία διανύσματα που υπάρχουν από το υποδέντρο του κόμβου.
- στους κόμβους φύλλα αποθηκεύονται και οι ακραίες τιμές τους στο πίνακα δεδομένων.

Τα τελικά αποτελέσματα του σχήματός μας που παρουσιάστηκαν και παραπάνω στα πειράματα δεν είναι ικανοποιητικά μάλλον για να λειτουργήσει ένα πραγματικό δίκτυο αισθητήρων. Αυτό μπορεί να οφείλεται στη μη καλή λειτουργία του σχήματος LSH πάνω σένα δίκτυο αισθητήρων όπως το χρησιμοποιήσαμε εμείς ή στην περαιτέρω ερεύνηση κάποιων ίσως λανθασμένων υποθέσεων που κάναμε.

6.5 Προτάσεις για περαιτέρω έρευνα

Περαιτέρω έρευνα πάνω στην εργασία αυτή θα μπορούσε να γίνει στη μελέτη εκ' νέου των τεχνικών που αναφέρονται παραπάνω για καλύτερη απόδοση τους ή στην υλοποίηση νέων όπως είναι η αποστολή μετρήσεων από κόμβο σε κόμβο οι οποίοι βρίσκονται στο ίδιο επίπεδο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] A. Andoni , P. Indyk , “Near-Optimal Hashing Algorithms For Approximate Nearest Neighbor in High Dimensions” focs, pp.459-468, 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), 2006

[2] A. Deligiannakis, Y. Kotidis, V. Vassalos, V. Stoumpos, A. Delis, "Another Outlier Bites the Dust: Computing Meaningful Aggregates in Sensor Networks," icde, pp.988-999, 2009 IEEE International Conference on Data Engineering, 2009

[3] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos , “Online outlier detection in sensor data using non-parametric models” , In VLDB '06: Proceedings of the 32nd international conference on Very large data bases (2006), pp. 187-198.

[4] Nikos Giatrakos, Yannis Kotidis, Antonios Deligiannakis, Vasilis Vassalos, Yannis Theodoridis , “TACO: tunable approximate computation of outliers in wireless sensor networks” , In Sigmod 2010

[5] X. Xiao, W. Peng, C. Hung, and W. Lee, "Using sensorranks for in-network detection of faulty readings in wireless sensor networks", in Proc. MobiDE, 2007, pp.1-8.

[6] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality", in Proc. STOC, 1998, pp.604-613.

[7] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions", in Proc. Symposium on Computational Geometry, 2004, pp.253-262.

[8] “SensorScope Wireless Distributed Sensing System for Environmental Monitoring” at: http://sensorscope.epfl.ch/index.php/Main_Page

[9] “CodeBlue: Wireless Sensors for Medical Care | Harvard Sensor Networks Lab” at: http://fiji.eecs.harvard.edu/CodeBlue

[10] M. Fisichella, F. Deng, and W. Nejdl, "Efficient Incremental Near Duplicate Detection Based on Locality Sensitive Hashing", in Proc. DEXA (1), 2010, pp.152-166.

[11] V.M. Zolotarev. One-Dimensional Stable Distributions. Vol.65 of Translations of Mathematical Monographs, American Mathematical Society, 1986.

[12] “Intel Lab Data” at: http://db.csail.mit.edu/labdata/labdata.html

[13] A. Andoni and P. Indyk, “E2LSH 0.1 user manual”, Technical report, Massachusetts Institute of technology (2004) http://web.mit.edu/andoni/www/LSH/manual.pdf

[14] “Flickr - Photo Sharing” at: http://www.flickr.com/

[15] “YouTube is a place to discover, watch, upload and share videos.” at: http://www.youtube.com/

[16] “TinyOS Home Page” at:http://www.tinyos.net/