ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
"ΠΛΗΡΟΦΟΡΙΚΗ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Data Storage for IoT data

Κωνσταντίνος Δ. Γεωργαντόπουλος

Επιβλέπων: **Αλέξανδρος Ντούλας,** Επίκουρος Καθηγητής

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2020

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Data Storage for IoT data

Κωνσταντίνος Δ. Γεωργαντόπουλος
Α.Μ.: CS2.18.0005

ΕΠΙΒΛΕΠΩΝ: **Αλέξανδρος Ντούλας,** Επίκουρος Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Αλέξης Δελής,** Καθηγητής
Μέμα Ρουσσοπούλου, Αναπληρώτρια Καθηγήτρια

Σεπτέμβριος 2020

ΠΕΡΙΛΗΨΗ

Σε αρκετές εφαρμογές Internet of Things (IoT) παρατηρούμε μία αφθονία από δεδομένα που έχουν τα εξής, κοινά χαρακτηριστικά: α) τα φορτία δεδομένων αυτά καθαυτά είναι σχετικά μικρά σε μέγεθος, β) βρίσκονται σε μία μορφή χρονικών σειρών, γ) οι μετρήσεις που παίρνουμε είναι πολύ συχνές, δ) οι μετρήσεις τυπικά επαναλαμβάνονται.

Τα δεδομένα IoT περιέχουν εξαιρετική πληροφορία σε ένα έξυπνο περιβάλλον που μπορεί να μας προσφέρει μια καλή διαίσθηση και γνώση πάνω στη λειτουργία του περιβάλλοντος καθώς και στη λειτουργικότητα των συσκευών. Οι περισσότερες έξυπνες συσκευές ανεβάζουν ολόκληρο το ρεύμα δεδομένων στο cloud για να το επεξεργαστούν και να τρέξουν διαφορετικών ειδών ερωτήματα. Αλλά, δεδομένου των χαρακτηριστικών των δεδομένων (π.χ. επαναλαμβανόμενα δεδομένα), μπορεί να υπάρξει σπατάλη πόρων.

Η πιο κοινή αναπαράσταση για την αποθήκευση και ανάλυση των IoT δεδομένων είναι η κλασική σχεσιακή αναπαράσταση με εγγραφές. Παρότι η αναπαράσταση αυτή εξυπηρετούσε τις ανάγκες των χρηστών για χρόνια, η όλο και αυξανόμενη ποσότητα των δεδομένων που παράγονται από αισθητήρες καθημερινά έχει φέρει στην επιφάνεια όλα τα αρνητικά στοιχεία της, όπως το στατικό σχήμα των δεδομένων, την αργή προσπέλαση ολόκληρων εγγραφών από τον δίσκο, τα ευρετήρια που απαιτούνται για την βελτιστοποίηση των ερωτημάτων και τη συνεχής ανάγκη για επανοργάνωση των δεδομένων όσο αυτά πληθαίνουν.

Για την αντιμετώπιση αυτών των μειονεκτημάτων έχουν υλοποιηθεί διάφορα συστήματα κατά καιρούς. Ένα από αυτά είναι το Συσχετιστικό σύστημα αναπαράστασης το οποίο είναι εξ ολοκλήρου βασισμένο στα δεδομένα και δεν χρειάζεται κανένα σχεδιασμό ή σχηματικό περιορισμό. Επίσης, η έλλειψη τέτοιων δομών σχεδιασμού το καθιστά και πιο οικονομικό από άποψη χώρου.

Σε αυτή την εργασία ο στόχος είναι η διερεύνηση και η υλοποίηση των διαφορετικών τεχνικών που αναφέρθηκαν για την αναπαράσταση και την μεταφορά IoT δεδομένων στο cloud με στόχο την εξοικονόμηση εύρους δικτύου και υπολογιστικών πόρων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Internet of Things

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: αποθήκευση, δεδομένα, χρονικές σειρές, συσχετιστικό, βάση δεδομένων

ABSTRACT

In several Internet-of-Things (IoT) applications we observe an abundance of data sharing the following characteristics: a) the individual data payloads are typically small in size b) they come in a time-series form c) the measurements we get are very frequent and d) the measurements are typically repeated.

IoT data capture great information in a smart space that can provide us with good insights and knowledge on the operation of the space, as well as the functionality of the devices. Most smart devices upload the whole stream of data to the cloud in order to process it and to run different types of queries. However, given the characteristics of the data (e.g. repeated data) there can be a waste of resources.

The most common representation for storage and analysis of IoT data is the classic relational, record-based approach. Although this approach satisfied users' needs for years, the rapid rise in size of data that are produced by sensors everyday has brought to light the drawbacks of this approach such as the static data model, the slow reading of whole records from the disc, the indexes needed for query optimization and the constant need to reorganize the data as they grow in size.

To address those drawbacks, alternative systems have been developed over the years. One such system is the correlation schema system which is exclusively dependent on data and there is no need for a predesigned schema or schematic constraints. Also the lack of a predesigned schema makes the system more efficient in storage usage.

In this project we plan to investigate and implement the different techniques mentioned earlier for representing and transferring IoT data to a cloud in order to save bandwidth and computational resources.

SUBJECT AREA: Internet of Things

KEYWORDS: storage, data, time series, correlation, database

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέπων, καθ. Αλέξανδρο Ντούλα, για την ευκαιρία που μου έδωσε για συνεργασία και την πολύτιμη συμβολή του στην ολοκλήρωση της παρούσας διπλωματικής εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία με τίτλο Data storage for IoT data εκπονήθηκε στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών. Η εκπόνηση της διπλωματικής εργασίας έγινε στα πλαίσια συνεργασίας με τον καθηγητή κ. Αλέξανδρο Ντούλα στον τομέα των Βάσεων Δεδομένων και του Internet of Things. Σκοπός της παρούσας εργασίας είναι η δημιουργία μιας πιο συμπαγούς μεθόδου αποθήκευσης των χρονικών δεδομένων και η απάντηση σε μια σειρά από ενδιαφέροντα ερωτήματα πάνω σε αυτή για μέτρηση της απόδοσης.

1. Εισαγωγή

Το Internet of Things (IoT) αναφέρεται σε ένα αρκετά μεγάλο αριθμό από "πράγματα" που είναι συνδεδεμένα στο διαδίκτυο έτσι ώστε να μπορούν να μοιράζονται δεδομένα με άλλα "πράγματα", όπως IoT εφαρμογές, συνδεδεμένες συσκευές, βιομηχανικά μηχανήματα κ.α. Όλες αυτές οι διασυνδεδεμένες συσκευές χρησιμοποιούν ενσωματωμένους αισθητήρες για να παράγουν δεδομένα και μερικές φορές για να δράσουν πάνω σε αυτά.

Το IoT υπόσχεται να βελτιώσει τον τρόπο ζωής και εργασίας μας. Μερικά από τα βασικά του πλεονεκτήματα είναι η λειτουργική αποδοτικότητα με αυτοματοποιημένες διαδικασίες και υψηλότερη παραγωγικότητα, τα καινοτόμα business plans με νέες ψηφιακές επιλογές και η βελτιωμένη εμπειρία για τους πελάτες με νέες και πιο διαδραστικές εμπειρίες αγορών.

Σήμερα οι συσκευές που βρίσκονται συνδεδεμένες στο IoT κυμαίνονται από καθημερινές συσκευές, όπως smartphones και smartwatches, μέχρι ολόκληρα συστήματα έξυπνων σπιτιών με εκατοντάδες διαφορετικού είδους και σκοπού αισθητήρες. Πρακτικά οτιδήποτε μπορεί να έχει έναν ενσωματωμένο αισθητήρα και να συνδεθεί στο διαδίκτυο. Σήμερα περίπου 127 συσκευές το δευτερόλεπτο συνδέονται στο διαδίκτυο ενώ μέχρι το 2025 το μέγεθος των IoT δεδομένων που παράγονται θα φτάνει τα 175 Zettabytes. [15]

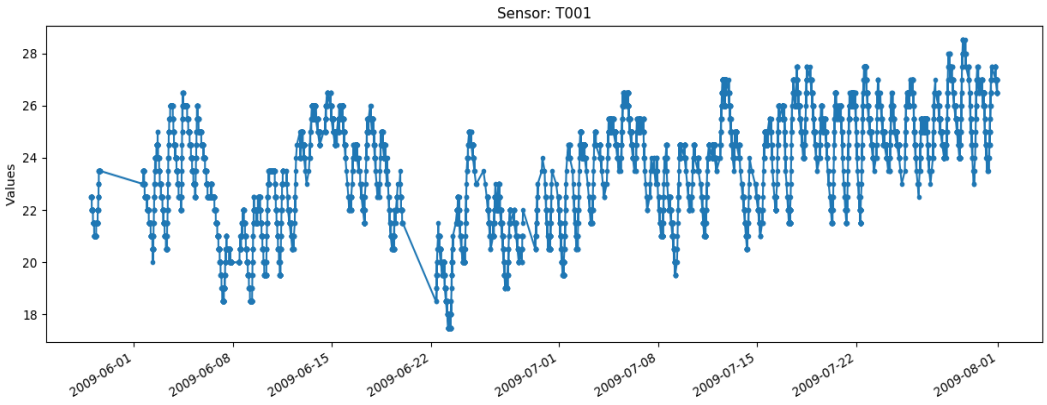
Ένα τυπικό smart home περιβάλλον, για παράδειγμα, αποτελείται από έναν σημαντικό αριθμό αισθητήρων (>5), είτε αριθμητικών είτε κατηγορικών, καθώς και από ένα μεγάλο διάστημα καταγραφών (>10 ημερών) για πιο αξιόπιστη μέτρηση της απόδοσης. Η μεγάλη γκάμα από αισθητήρες αναδεικνύει με τον καλύτερο τρόπο όλα εκείνα τα χαρακτηριστικά των χρονικών δεδομένων που θέλουμε να συμπιέσουμε: επανάληψη και περιοδικότητα. Για παράδειγμα όπως φαίνεται και στο Σχήμα 1 ένας αισθητήρας θερμοκρασίας στο σαλόνι θα δείχνει πολλές φορές την τιμή της θερμοκρασίας δωματίου (24-27 βαθμούς κελσίου) και σπάνια θα ξεμακραίνει από αυτό το εύρος.

Μέχρι σήμερα η πιο γνωστή τεχνική για την αποθήκευση και μετέπειτα ανάκτηση δεδομένων ήταν η σχεσιακή αναπαράσταση. Σε αυτήν όλα τα δεδομένα χαρακτηρίζονται από ένα προκαθορισμένο σχήμα το οποίο αποτελείται από πίνακες που περιέχουν εγγραφές. Για την αναφορά κάθε εγγραφής χρησιμοποιείται η φυσική θέση της εγγραφής στο δίσκο και κάθε εγγραφή πρέπει να μεταφέρεται αυτούσια σαν μονάδα. Έτσι αν χρειάζεται να γίνει ανάκτηση κάποιου συγκεκριμένου πεδίου τότε θα

πρέπει να διαβαστεί ολόκληρη η εγγραφή, ενώ για την τροποποίηση χρειάζεται να ξαναγραφεί ολόκληρη η εγγραφή.

Αυτή η διαδικασία είναι ιδιαίτερα χρονοβόρα και καθόλου αποδοτική όσον αφορά λειτουργίες εισόδου/εξόδου στον δίσκο. Λαμβάνοντας υπόψη και την ραγδαία αύξηση των δεδομένων που προαναφέραμε το σύστημα καθίσταται αρκετά αναποτελεσματικό.

Για να αντιμετωπιστούν τέτοια προβλήματα έχουν υλοποιηθεί κατά καιρούς διαφορετικά συστήματα αποθήκευσης με ξεχωριστά χαρακτηριστικά. Ένα από αυτά είναι το συσχετιστικό σύστημα βάσεων δεδομένων το οποίο χρησιμοποιεί μια δομή βασισμένη στις τιμές (Value Based Storage - VBS). Με αυτόν τον τρόπο προσπερνάει το πρόβλημα δεικτοδότησης της δομής εγγραφών αλλά και προσφέρει νέα χαρακτηριστικά εξερεύνησης και ανάλυσης των δεδομένων. Το μοντέλο VBS αποθηκεύει κάθε ξεχωριστή τιμή μόνο μία φορά και μετά χρησιμοποιεί μεταδεδομένα για να κρατήσει την μορφή των εγγραφών στο δίσκο ελαχιστοποιώντας το εύρος δικτύου που χρησιμοποιείται καθώς τα πραγματικά αποθηκευμένα δεδομένα είναι τόσο μικρά που ελαχιστοποιούν τις προσπελάσεις του δίσκου.



Σχήμα 1 Σχεδιάγραμμα ενός αισθητήρα θερμοκρασίας σαλονιού.

Στη συνέχεια της εργασίας μελετάμε στο κεφάλαιο 2 συγκρίνουμε την σχεσιακή προσέγγιση με το συσχετιστικό σύστημα και παρουσιάζουμε και το κατά-Στήλη σύστημα για πληρότητα και στο κεφάλαιο 3 παρουσιάζουμε την πειραματική αξιολόγηση των 2 συστημάτων με χρήση διάφορων ερωτημάτων και τέλος παρουσιάζουμε μερικές τεχνικές για εντοπισμό δυσλειτουργιών σε αισθητήρες.

2. Αποθήκευση Δεδομένων

2.1 Σχεσιακό Σχήμα Δεδομένων

Πριν από την εκτέλεση των ερωτημάτων πάνω στα εκάστοτε δεδομένα, θα πρέπει να ορίσουμε ένα ενιαίο λογικό σχήμα που να χαρακτηρίζει όλα τα δεδομένα μας. Η πιο συχνά χρησιμοποιούμενη μορφή αποθήκευσης είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων. Υπάρχουν πολλές εμπορικές εκδόσεις ενός τέτοιου συστήματος στην αγορά, όπως η MySQL, και όλες υποστηρίζουν το κοινό χαρακτηριστικό της αποθήκευσης σχετιζόμενων τιμών σε μια φυσικά συνεχή εγγραφή. Αυτή η μορφή ήταν η κυρίαρχη στην αγορά για χρόνια λόγω των πλεονεκτημάτων της στην επεξεργασία και αποθήκευση συναλλαγών και τη δημιουργία αναφορών.

Πίνακας 1: Πειραματικά δεδομένα για επίδειξη σχεσιακής αναπαράστασης

Timestamp	Sensor	Value	Activity
22-06-2015 22:32:06	M41	0.5	Activity1
22-06-2015 22:32:08	T01	25	Activity1
22-06-2015 22:32:25	M41	0.25	Activity1
22-06-2015 22:32:30	T02	27	Activity1

Αρχικά, επειδή έχουμε να κάνουμε με χρονολογικές σειρές δεδομένων, σίγουρα θα χρειαστούμε μία στήλη *Timestamp* τύπου *DateTime* της MySQL. Έπειτα κάθε σετ δεδομένων αποτελείται από ένα σύνολο από αισθητήρες πράγμα που καθιστά την ύπαρξη μιας στήλης *Sensor* με το όνομα κάθε αισθητήρα απαραίτητη. Ακόμη, κάθε αισθητήρας παίρνει και διάφορες τιμές με την πάροδο του χρόνου. Αυτές οι τιμές μπορεί να είναι είτε αριθμητικές είτε αλφαριθμητικές, επομένως θα χρειαστούμε μία στήλη *Value* γενικού τύπου. Τέλος, έχουμε και κάποια annotated σύνολα δεδομένων τα οποία μπορεί να προσδιορίζουν την δραστηριότητα που εκτελούσε ένα άτομο σε κάποια χρονική σήμανση. Αυτές οι πληροφορίες είναι συμπληρωμένες χειροκίνητα και δεν

αφορούν όλο το χρονικό εύρος του πειράματος. Επομένως θα χρειαστούμε και 2 στήλες επιπλέον, τις *Activity* και *Tag*, οι οποίες θα έχουν προαιρετικό χαρακτήρα.

Τελικά το λογικό σχήμα της σχεσιακής βάσης μας φαίνεται στον Πίνακα 1 μαζί με κάποια πειραματικά δεδομένα.

Παρά την κοινή αποδοχή του ανά τα χρόνια και την εκτενή χρήση του, το σχεσιακό σχήμα έχει κάποιους περιορισμούς που υπονομεύουν την χρησιμότητά του. Από τους πιο σοβαρούς είναι το πρόβλημα της κατάλληλης δεικτοδότησης των δεδομένων έτσι ώστε να μπορεί να ανταποκρίνεται γρήγορα σε ένα αναλυτικό περιβάλλον. Επίσης η σχεσιακή δομή υποφέρει από προβλήματα απόδοσης λόγω της ανάγκης για μεταφορά ολόκληρων εγγραφών σε κάθε λειτουργία ανάγνωσης ή εγγραφής.

Ακόμη και με την εισαγωγή ευρετηρίων για πιο γρήγορη προσπέλαση στα διάφορα πεδία ενός πίνακα μπορούν να προκύψουν άλλα προβλήματα. Όταν ο αριθμός των πεδίων είναι μικρός, π.χ. 1-2 πεδία, τότε μπορούν να κατασκευαστούν 2 ευρετήρια χωρίς σημαντική πτώση της απόδοσης. Όταν όμως τα δεδομένα μας χαρακτηρίζονται από πολλά πεδία ή είναι μεγάλα σε πλήθος τότε η κατασκευή αλλά και ενημέρωση τόσων ευρετηρίων για έναν πίνακα εκτοξεύει τόσο τον χώρο αποθήκευσης όσο και τον χρόνο ενημέρωσης των δεδομένων μας.

Ένα δεύτερο πρόβλημα με το σχεσιακό σύστημα είναι η βελτιστοποίηση των ερωτημάτων που δέχεται με τη χρήση μόνο πεδίων ευρετηρίου όταν εμπλέκονται πάρα πολλά ευρετήρια στο ερώτημα. Ενώ το "πάρα πολλά" είναι ένας σχετικός όρος χωρίς καθορισμένη τιμή, κάθε σχεσιακό ΣΔΒΔ έχει ένα πρακτικό όριο. Συνεπώς, κανείς δεν μπορεί, σε λογικά πλαίσια, να περιλαμβάνει αρκετά ευρετήρια για την πλήρη υποστήριξη περίπλοκων ερωτημάτων και αναλυτικών διεργασιών.

Τρίτον ένα σχεσιακό σύστημα περιορίζεται από το αρχικά καθορισμένο σχήμα της βάσης. Με άλλα λόγια αν χρειαστεί να γίνει κάποια αλλαγή στο σχήμα στο μέλλον, π.χ. εισαγωγή νέας στήλης σε πίνακα, τότε θα πρέπει είτε ολόκληρος ο πίνακας να ξαναγραφεί από την αρχή με την νέα στήλη στη θέση της, είτε η νέα στήλη να γραφτεί σε ξεχωριστό σημείο με ειδική δεικτοδότηση.

Τέλος ένα ακόμη πρόβλημα εμφανίζεται με την συνεχή εισαγωγή/επεξεργασία/διαγραφή δεδομένων στη βάση. Επειδή το σχήμα έχει σταθερή δομή τότε οι συνεχείς αλλαγές μπορεί να αφήνουν κενά ενδιάμεσα μπλοκ στο δίσκο, διάσπαρτες εγγραφές κ.α. Για να αντιμετωπιστεί αυτό μια σχεσιακή βάση θα πρέπει να αναδιοργανώνεται ανά τακτά χρονικά διαστήματα για να κρατάει τη δομή της αποδοτική. Αυτή η διαδικασία απαιτεί

σημαντικούς υπολογιστικούς πόρους καθώς και τη βάση να είναι μη διαθέσιμη στο δίκτυο καθιστώντας την απροσπέλαστη στους αναλυτές για ένα σεβαστό χρονικό διάστημα.

2.2 Κατά-Στήλη Βάση Δεδομένων

Μία κατά στήλη βάση δεδομένων αποθηκεύει τα δεδομένα με πολύ διαφορετικό τρόπο από ότι η σχεσιακή. Σε αυτή τη δομή, τα δεδομένα αποθηκεύονται σε σύνολα κατά στήλη, δηλαδή όλες οι τιμές την στήλης 1 αποθηκεύονται σε ένα σύνολο, της στήλης 2 σε άλλο σύνολο κ.ο.κ. Επιπρόσθετα με τις τιμές, όλη η πληροφορία που χρειάζεται για την ανακατασκευή των εγγραφών με τις σωστές θέσεις βρίσκεται μέσα σε κάθε σύνολο.

Αυτή η δομή παρέχει σημαντικά πλεονεκτήματα σε αναλυτικά περιβάλλοντα και χρησιμοποιεί λιγότερο χώρο στο δίσκο σε σχέση με τα παραδοσιακά σχεσιακά ΣΔΒΔ.

Το πιο προφανές πλεονέκτημα της δομής αυτής είναι ότι κατά την επιλογή εγγραφών σε ένα ερώτημα, είναι αναγκαία μόνο η προσπέλαση των στηλών που περιλαμβάνονται ως κριτήρια ή αποτελέσματα σε αυτό. Κάτι τέτοιο μπορεί να κάνει το χρόνο απάντησης του ερωτήματος έως και 10 φορές ταχύτερο από ότι σε ένα σχεσιακό σύστημα. Αν οι στήλες είναι αποθηκευμένες σε ταξινομημένη μορφή, τότε η βελτίωση του χρόνου απάντησης μπορεί να γίνει ακόμα μεγαλύτερη.

Το μεγαλύτερο μειονέκτημα της δομής αυτής είναι κατά την εισαγωγή εγγραφών ή την επεξεργασία κατά συστάδες (batch processing). Για την εισαγωγή μιας νέας εγγραφής χρειάζεται το διάβασμα ενός συνόλου στήλης για κάθε στήλη της εγγραφής, η απόφαση για το πού θα πρέπει να τοποθετηθεί μία τιμή στο σετ, η εισαγωγή της τιμής αυτής στη σωστή θέση και τέλος η κατασκευή της απαραίτητης πληροφορίας σύνδεσης έτσι ώστε να είναι δυνατή η ανάκτηση της εγγραφής και η εισαγωγή της στην κατάλληλη μορφή εγγραφής.

Αυτό δεν αποτελεί σοβαρό πρόβλημα αν οι τιμές αποθηκεύονται σε κάθε σετ με τη σειρά με την οποία εισάγονται και το μόνο που χρειάζεται είναι η τοποθέτησή τους στο τέλος του συνόλου. Βέβαια, χρησιμοποιώντας αυτή τη μέθοδο, η διαδικασία επιλογής και ανάκτησης εγγραφών θα είναι πιο αργές καθώς κάθε στήλη θα πρέπει πάντα να διαβάζεται από την αρχή ως το τέλος.

Παρόλα αυτά, αν οι στήλες αποθηκεύονται ταξινομημένες, η εισαγωγή μιας νέας εγγραφής είναι χρονοβόρα διαδικασία καθώς χρειάζεται η ταξινόμηση και αναδιοργάνωση των τιμών της στήλης. Παρότι η μέθοδος με ταξινόμηση βελτιστοποιεί την διαδικασία επιλογής και ανάκτησης, χρειάζεται συνεχή ρύθμιση για να παρέχει

βέλτιστη απόδοση και να ικανοποιεί τις ανάγκες της αγοράς. Καθώς τα θέματα ρύθμισης απόδοσης διαφέρουν, συχνά είναι τόσο εκτενή όσο σε ένα σχεσιακό ΣΔΒΔ.

2.3 Συσχετιστική Βάση Δεδομένων

Μία συσχετιστική βάση δεδομένων (Correlation Database) είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) που είναι ανεξάρτητο του μοντέλου δεδομένων. Σε αντίθεση με τα σχεσιακά ΣΔΒΔ, που ακολουθούν μία κατά πλειάδα προσέγγιση ή τα κατά στήλη ΣΔΒΔ, που χρησιμοποιούν μία κατά στήλη προσέγγιση, μία συσχετιστική ΒΔ χρησιμοποιεί μία βασισμένη στην τιμή προσέγγιση αποθήκευσης (Value Based Storage - VBS). Σε αυτή την προσέγγιση κάθε μοναδική τιμή δεδομένων αποθηκεύεται μόνο μία φορά και ένα αυτόματα παραγόμενο σύστημα δεικτοδότησης κρατάει την πληροφορία για όλες τις τιμές. Ακριβώς επειδή ένα συσχετιστικό ΣΔΒΔ αποθηκεύει κάθε τιμή δεδομένων μόνο μία φορά, το πραγματικό μέγεθος της βάσης είναι αρκετά μικρότερο από τα παραδοσιακά ΣΔΒΔ και χωρίς την χρήση τεχνικών συμπίεσης δεδομένων. Επειδή τα δεδομένα επαναλαμβάνονται, σε γενικές γραμμές οι πραγματικές τιμές δεδομένων που αποθηκεύονται από το VBS είναι κατά πολύ λιγότερες από τον αριθμό των εγγραφών. Έτσι, το VBS ελαχιστοποιεί την κατανάλωση bandwidth, καθώς το σύνολο των τιμών δεδομένων μπορούν να παραμένουν συχνά στη μνήμη επηρεάζοντας το I/O εύρος μόνο κατά την φόρτωση των προς αναζήτηση εγγραφών.

Το μοντέλο VBS που ακολουθείται από ένα συσχετιστικό ΣΔΒΔ αποτελείται από 3 βασικά μέρη:

- Ένα λεξικό δεδομένων (Μεταδεδομένα)
- Ένα σύνολο δεδομένων δεικτοδότησης και σύνδεσης
- Τις πραγματικές τιμές δεδομένων που αποτελούν την αποθηκευμένη πληροφορία

Οι τιμές δεδομένων αποθηκεύονται μόνο μία φορά και σε ταξινομημένα σύνολα, όλοι οι ακέραιοι σε ένα σύνολο, όλες οι συμβολοσειρές μαζί σε άλλο κλπ. Επίσης, μαζί με τις πραγματικές τιμές δεδομένων αποθηκεύονται και πληροφορίες για τις σχέσεις μεταξύ τους, παρόμοια διαδικασία με τα ξένα κλειδιά στις παραδοσιακές βάσεις δεδομένων, αλλά η σχέση είναι γνωστή στο λεξικό και αποθηκευμένη σαν τιμή πράγμα που κάνει την μετάβαση μεταξύ των πινάκων αυτόματη. Επιπρόσθετα, από τη στιγμή που υπάρχει μόνο ένα μέρος για αναζήτηση όταν επιλέγουμε εγγραφές, όλες οι αναζητήσεις παίρνουν το πλεονέκτημα της ταχύτητας της δεικτοδότησης.

Επίσης με τη δομή VBS που χρησιμοποιείται σε ένα συσχετιστικό ΣΔΒΔ δεν χρειάζεται καμία σχεδιαστική απόφαση και οι εξελισσόμενες εργασιακές απαιτήσεις δεν αναμένεται να αναγκάσουν σε αλλαγή τη φυσική δομή. [4][5]

Για παράδειγμα αν υποθέσουμε ότι έχουμε τα δεδομένα του πίνακα 1 τότε η αναπαράσταση των τιμών που εμφανίζονται στον πίνακα σε ένα συσχετιστικό ΣΔΒΔ θα ήταν αυτή που φαίνεται στον πίνακα 2. Η αναπαράσταση των εγγραφών ολοκληρωμένων με δείκτες φαίνεται στον πίνακα 3. Ενδιαφέρον παρουσιάζει η τιμή "Activity1" η οποία, αν και εμφανίζεται πολλές φορές στα πραγματικά δεδομένα, εν τέλει αποθηκεύεται μόνο μία φορά. Αυτό το πλεονέκτημα αυξάνει όσο η επανάληψη των τιμών μεγαλώνει στα πραγματικά δεδομένα.

Πίνακας 2 Αναπαράσταση δεδομένων σε συσχετιστικό ΣΔΒΔ

ID	Value
1	22-06-2015 22:32:06
2	22-06-2015 22:32:08
3	22-06-2015 22:32:25
4	22-06-2015 22:32:30
5	0.25
6	0.5
7	25
8	27
9	M41
10	T01
11	T02
12	Activity1

Στην VBS δομή κάθε τιμή αποθηκεύεται μόνο μία φορά και της δίνεται ένα μοναδικό αναγνωριστικό ξέχωρα από την θέση της στα πραγματικά δεδομένα. Έπειτα τα πραγματικά δεδομένα ανασυγκροτούνται με αναφορές πάνω σε αυτά τα αναγνωριστικά.

Με το VBS η απόδοση των ερωτημάτων είναι εξαιρετική καθώς για την προετοιμασία του ερωτήματος δεν διαβάζεται ολόκληρη εγγραφή. Αντίθετα, η διαδικασία αναζήτησης πραγματοποιείται μέσα από ένα σύνολο τιμών που είναι πάντα ταξινομημένο και έναν

αλγόριθμο δεικτοδότησης που αναγνωρίζει τις κατάλληλες εγγραφές. Τα βήματα αυτά χρειάζονται ελάχιστο I/O φόρτο και κατά συνέπεια εκτελούνται αρκετά γρήγορα.

Σε ένα συσχετιστικό ΣΔΒΔ η απόδοση φόρτωσης είναι σημαντικά καλύτερη από ότι στα σχεσιακά ΣΔΒΔ. Από την στιγμή που η πλειοψηφία των μοναδικών τιμών έχει φορτωθεί στη μνήμη οι εισαγωγές νέων εγγραφών απαιτούν μόνο την ενημέρωση του συνόλου δεικτοδότησης το οποίο είναι αρκετά γρήγορο. Επίσης και οι διαγραφές εγγραφών δεν επηρεάζουν τα αποθηκευμένα δεδομένα, απαιτώντας μόνο διαγραφές στη δομή δεικτοδότησης.

Η απόδοση επίσης αυξάνεται από την εγγενή συμπίεση που προσφέρει το VBS στα δεδομένα, καθώς κάθε μοναδική τιμή αποθηκεύεται μόνο μια φορά. Το πλήθος των δεδομένων περιορίζεται σε ένα μικρό κλάσμα της αντίστοιχης σχεσιακής δομής χωρίς την ανάγκη χρήσης επιπλέον αλγορίθμων συμπίεσης.

Ακόμη, καθώς η βάση μεγαλώνει και το σύνολο τιμών γίνεται πιο περίπλοκο, το σύνολο δεικτοδότησης αυξάνει σε σταθερό ρυθμό με τα δεδομένα, αυξάνοντας μεν το μέγεθος της βάσης, αλλά κατά πολύ μικρό ποσοστό σε σχέση με το καθαρό σύνολο δεδομένων. Με ένα πλήθος 100 εκατομμυρίων εγγραφών αποθηκευμένων, ένα συσχετιστικό ΣΔΒΔ έχει περίπου το ίδιο μέγεθος με τα καθαρά αυθεντικά δεδομένα. Καθώς η βάση μεγαλώνει περαιτέρω το μέγεθος του Συσχετιστικού ΣΔΒΔ σε σχέση με τα καθαρά δεδομένα μικραίνει. Ακόμη και με τα δεδομένα μας κανονικοποιημένα, π.χ. σε 3NF όπου μειώνονται αρκετά τα διπλότυπα και η επανάληψη πληροφορίας, το συσχετιστικό σχήμα υπερέχει [14][16].

Ένα τυπικό σχήμα σε 3NF θα έχει έναν πίνακα με μεγάλο αριθμό εγγραφών, μερικούς πίνακες με πολλές εγγραφές και αρκετούς πίνακες με λίγες εγγραφές. Στον πίνακα 3 έχουμε το πιθανό σχήμα περίπου 1 terabyte δεδομένων παραγγελιών προϊόντων από πελάτες οργανωμένο σε 4 πίνακες σε μορφή 3NF.

Μία λειτουργική απαίτηση όπως, "δείξε μου όλους τους πελάτες που αγόρασαν το προϊόν Α" θα οδηγούσε σε ένα SQL ερώτημα της μορφής:

```
SELECT name, address, city, state, ZIP
 FROM customer,  product,  transaction
WHERE customer.customer_id = transaction.customer_id AND transaction.product_id =
product.product_id AND product.name = "Product A";
```

Για σύγκριση, ας θεωρήσουμε ως επιλογή ενός προϊόντος που αγοράστηκε 1000 φορές από 500 πελάτες.

Αγνοώντας τη δομή των μπλοκ του δίσκου και υποθέτοντας σωστά ευρετήρια, το σχεσιακό ΣΔΒΔ θα χρειαζόταν να προσπελάσει 1 εγγραφή προϊόντος, 1000 εγγραφές λεπτομέρειας συναλλαγής, περίπου 1000 εγγραφές παραγγελιών και 500 εγγραφές πελατών για να απαντήσει το ερώτημα. Αυτό συνεπάγεται σε μία μεταφορά περισσότερων από 550.000 bytes δεδομένων για την ανάκτηση περίπου 50.000 bytes πληροφορίας.

Αντίστοιχα το συσχετιστικό ΣΔΒΔ, πάλι αγνοώντας τη δομή των μπλοκ, θα χρειαζόταν η προσπέλαση μίας τιμής από τη στήλη όνομα του προϊόντος και μετά η ανάκτηση του ID της τιμής αυτής. Το ID αυτό θα χρησιμοποιούταν για την εύρεση 1000 συναλλαγών που με τη σειρά τους θα χρησιμοποιούνταν για την εύρεση των IDs των παραγγελιών. Αυτά θα χρησιμοποιούνταν για την ανάκτηση των εγγραφών παραγγελίας και η διαδικασία θα επαναλαμβανόταν για την εύρεση των εγγραφών των πελατών. Τελικά το ερώτημα θα απαντιόταν με την επιστροφή των πληροφοριών πελάτη. Αυτή η διαδικασία, ενώ αυξάνει σημαντικά τον φόρτο CPU ελαχιστοποιεί τις μεταφορές I/O. Συγκεκριμένα 60.000 bytes προσπελαύνονται για την ανάκτηση 50.000 bytes πληροφορίας.[14]

Πίνακας 3 Μέγεθος δεδομένων με οργάνωση 3NF

Table	Records	Columns	Gigabytes
Transaction	1.500.000.000	20	720
Order	100.000.000	75	180
Customer	5.000.000	1000	120
Product	25.000	75	0.045

Τέλος σε ένα συσχετιστικό ΣΔΒΔ δεν υπάρχουν ευρετήρια πινάκων. Συνεπώς το συνολικό μέγεθος της βάσης δεν μεγαλώνει με την κατασκευή ευρετηρίων. Αυτό έχει το διπλό πλεονέκτημα της εξοικονόμησης χώρου στο δίσκο και της επιτάχυνσης των διαδικασιών φόρτωσης.

Πίνακας 4 Αναπαράσταση εγγραφών σε συσχετιστικό ΣΔΒΔ

Timestamp	Sensor	Value	Activity
1	9	6	12
2	10	7	12
3	9	5	12
4	11	8	12

3. Πειραματική Αξιολόγηση

Η προσομοίωση και η εκτέλεση των ερωτημάτων για το σχεσιακό σχήμα έγινε με το σύστημα της MySQL. Εφόσον δεν βρήκαμε κάποιο open source συσχετιστικό ΣΔΒΔ, θα το προσομοιώσουμε χρησιμοποιώντας πίνακες με auto incremented Ids στο column store περιβάλλον της MonetDB που έρχεται πιο κοντά -σε μορφή και χαρακτηριστικά- με ένα συσχετιστικό ΣΔΒΔ. Για την αποθήκευση των δεδομένων δεν χρησιμοποιήθηκε ένας ενιαίος πίνακας όπως αναφέρει η βιβλιογραφία αλλά ένας πίνακας για κάθε στήλη (Sensor, Value κλπ). Αυτό γίνεται γιατί χρησιμοποιούμε σύστημα που δεν είναι εγγενώς συσχετιστικό, με ό,τι αυτό συνεπάγεται στην δεικτοδότηση των δεδομένων όπως προαναφέραμε, και θέλουμε να διατηρούμε κάθε διαφορετικό σύνολο δεδομένων οργανωμένο για εύκολη πρόσβαση. Το διάβασμα και η προεπεξεργασία των δεδομένων πριν την τελική αποθήκευσή τους έγινε σε περιβάλλον Python με χρήση της βιβλιοθήκης pandas. [6], [7]

Για την υλοποίηση και την αξιολόγηση των μεθόδων συμπαγούς αναπαράστασης των χρονικών σειρών, χρησιμοποιούμε μία συλλογή από 4 datasets που αφορούν δεδομένα χρήσης πάνω σε καθημερινά αντικείμενα, ασχολίες και κινήσεις.

Τα 2 πρώτα datasets αποτελούν μια συλλογή από γεγονότα αισθητήρων που συλλέχθηκαν σε διάστημα 3 μηνών στο έξυπνο διαμέρισμα WSU του CASAS κατά την άνοιξη και το καλοκαίρι, αντίστοιχα, του 2009. Το διαμέρισμα φιλοξένησε 2 κατοίκους οι οποίοι τηρούσαν τις συνηθισμένες καθημερινές τους δραστηριότητες που καταγράφονταν από 86 αισθητήρες διαφορετικών τύπων και θέσεων για ένα διάστημα 2 μηνών. Οι τυπικές δραστηριότητες αφορούσαν τη χρήση του τηλεφώνου, πλύσιμο χεριών, προετοιμασία φαγητού, φαγητό, περίθαλψη και καθαριότητα. Το διαμέρισμα αποτελούταν από 3 κρεβατοκάμαρες, ένα μπάνιο, μια κουζίνα και ένα σαλόνι. Επίσης στο διαμέρισμα είχαν τοποθετηθεί αισθητήρες κίνησης ανά περίπου 1 μέτρο μεταξύ τους. Ακόμη είχαν τοποθετηθεί ψηφιακοί αισθητήρες θερμοκρασίας και αναλογικοί αισθητήρες για μέτρηση ζεστού/κρύου νερού και χρήση φούρνου ή θερμαντικού [1],[2].

Το τρίτο και τέταρτο dataset ανήκουν στην οικογένεια του CityPulse EU FP7 project. Πρόκειται για ένα dataset με καταγραφές κίνησης οχημάτων (μέση ταχύτητα, αριθμός οχημάτων κ.α.) στην πόλη Aarhus της Δανίας κατά την περίοδο Φεβρουαρίου-Ιουνίου 2014 με 449 σημεία παρατήρησης συνολικά. Κάθε μέτρηση σε ένα σημείο παρατήρησης έχει απόσταση 5 λεπτών από την προηγούμενη και περιέχει τη μέση ταχύτητα των οχημάτων που πέρασαν σε αυτά τα 5 λεπτά, το μέσο χρόνο παρατήρησης οχημάτων και τον αριθμό οχημάτων που πέρασαν από το σημείο του

δρόμου καθώς και το id του σημείου αυτού. Ως αισθητήρες θεωρούμε τις καταγραφές ταχύτητας και τον αριθμό οχημάτων με τιμές τις αντίστοιχες που καταγράφονται από κάποιο σημείο παρατήρησης σε δεδομένο χρονικό σημείο.

Το τελευταίο dataset αποτελείται από δεδομένα ρύπανσης για την ίδια πόλη κατά την περίοδο Αυγούστου-Οκτωβρίου 2014. Το dataset αυτό ουσιαστικά αποτελεί συμπλήρωμα του προηγούμενου και παρέχει στοιχεία μόλυνσης για κάθε σημείο παρατήρησης του δρόμου που υπάρχει και στο άλλο. Έχουμε ξανά 449 σημεία παρατήρησης συνολικά με μετρήσεις ανά 5 λεπτά και για κάθε μία έχουμε τιμές για όζον, μονοξείδιο του άνθρακα, διοξείδιο του θείου, διοξείδιο του αζώτου και αιωρούμενα σωματίδια. Θεωρούμε ως αισθητήρες για την μελέτη μας κάθε στοιχείο μόλυνσης ξεχωριστά και ως τιμές τους τις τιμές ρύπανσης που καταγράφονται από κάποιο σημείο παρατήρησης σε δεδομένο χρονικό σημείο. [8],[9],[10]

Οι μετρήσεις έγιναν σε σύστημα με **CPU**: Intel Core i5-4570@3.2GHz με χρήση έως 4 threads όπου χρειάζεται, **RAM**: 8GB, **Disk:** SSD 128GB. Για τα ερωτήματα που ακολουθούν θα θεωρήσουμε ως ενδεικτικό dataset τους πίνακες 1 και 3 που προηγήθηκαν. Οι πίνακες αυτοί θα θεωρήσουμε ότι έχουν το όνομα Events.

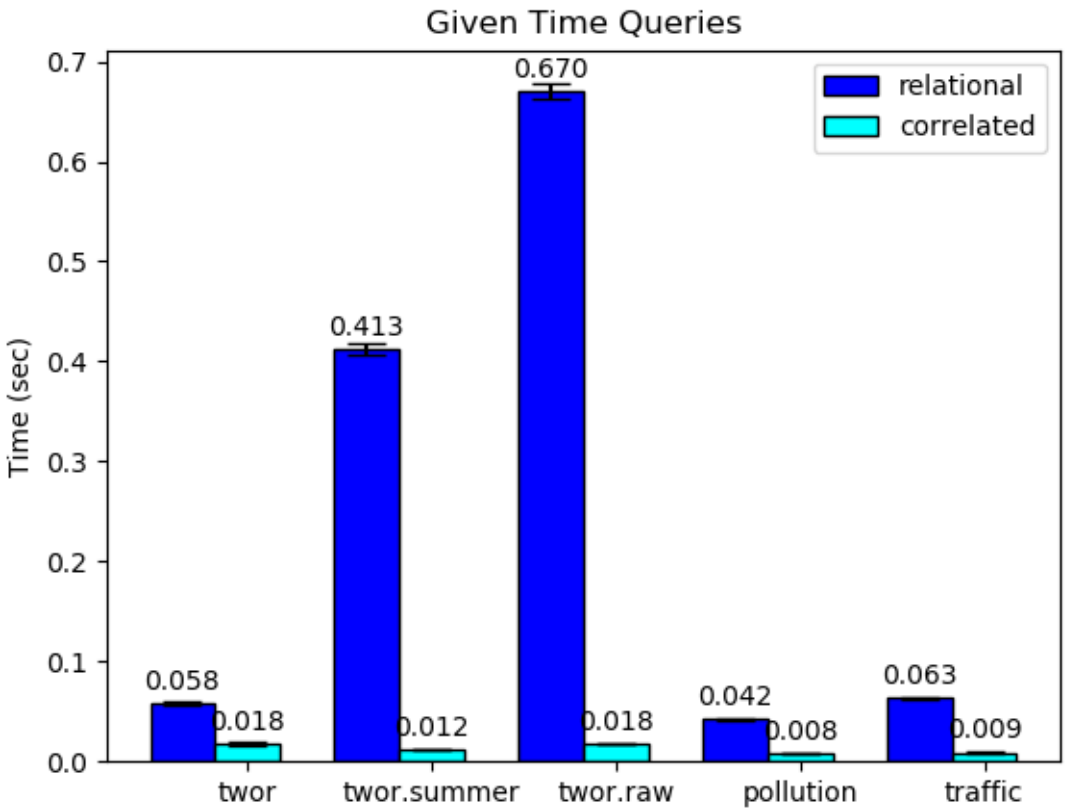
3.1 Τιμή αισθητήρα σε συγκεκριμένη χρονική στιγμή

Μια αρκετά βασική αναζήτηση σε χρονικά δεδομένα είναι η τιμή που έχει ένας αισθητήρας μια συγκεκριμένη χρονική στιγμή. Για παράδειγμα αυτό μπορεί να είναι χρήσιμο όταν θέλουμε να δούμε τις μετρήσεις ενός αισθητήρα ανά ώρα ή όταν θέλουμε να δούμε την θερμοκρασία ενός δωματίου μια συγκεκριμένη στιγμή. Πρόκειται για ένα αρκετά απλό ερώτημα. Η μοναδική "πρόκληση" που μπορεί να υπάρχει είναι όταν η χρονική στιγμή που δίνεται σαν παράμετρος δεν υπάρχει ακριβώς στα δεδομένα μας και πρέπει να βρεθεί η τιμή στην αμέσως πιο κοντινή χρονική στιγμή. Τα ενδεικτικά SQL ερωτήματα φαίνονται στον πίνακα 5.

Πίνακας 5 SQL Ερωτήματα τιμής σε συγκεκριμένη χρονική στιγμή

Relational	*SELECT e.Value FROM Events e WHERE e.Sensor = :sensor AND e.Timestamp <= :time ORDER BY e.Timestamp DESC LIMIT 1;*"
Correlation	*SELECT v.value FROM Events e, Sensor s, Value v WHERE e.sensor_id = s.id AND s.sensor_name = :sensor AND e.value_id = v.id AND e.Timestamp <= :time ORDER BY e.Timestamp DESC LIMIT 1;*

όπου :sensor και :time υποδηλώνουν τις παραμέτρους για τον αισθητήρα και την χρονική στιγμή αντίστοιχα. Για παράδειγμα αν θεωρήσουμε ως :sensor τον T01 και ως :time το 22-06-2015 22:32:10 τότε θα πάρουμε ως αποτέλεσμα την τιμή 25.



Σχήμα 2 Συγκριτικοί χρόνοι ερωτήματος τιμής σε συγκεκριμένη χρονική στιγμή

Στο σχήμα 2 βλέπουμε τους μέσους χρόνους εκτέλεσης των παραπάνω ερωτημάτων για κάθε ένα από τα datasets. Παρατηρούμε ότι για όλα τα datasets οι χρόνοι εκτέλεσης των ερωτημάτων του συσχετιστικού σχήματος είναι αρκετά μικρότεροι. Στον κατακόρυφο άξονα έχουμε τον χρόνο σε δευτερόλεπτα και στον οριζόντιο τα διαφορετικά datasets. Οι γραμμές εμπιστοσύνης που φαίνονται στην κορυφή κάθε μπάρας μας δείχνουν το εύρος των χρόνων όλων των επαναλήψεων των ερωτημάτων.

3.2 Μέση τιμή αισθητήρα σε μια δεδομένη χρονική περίοδο

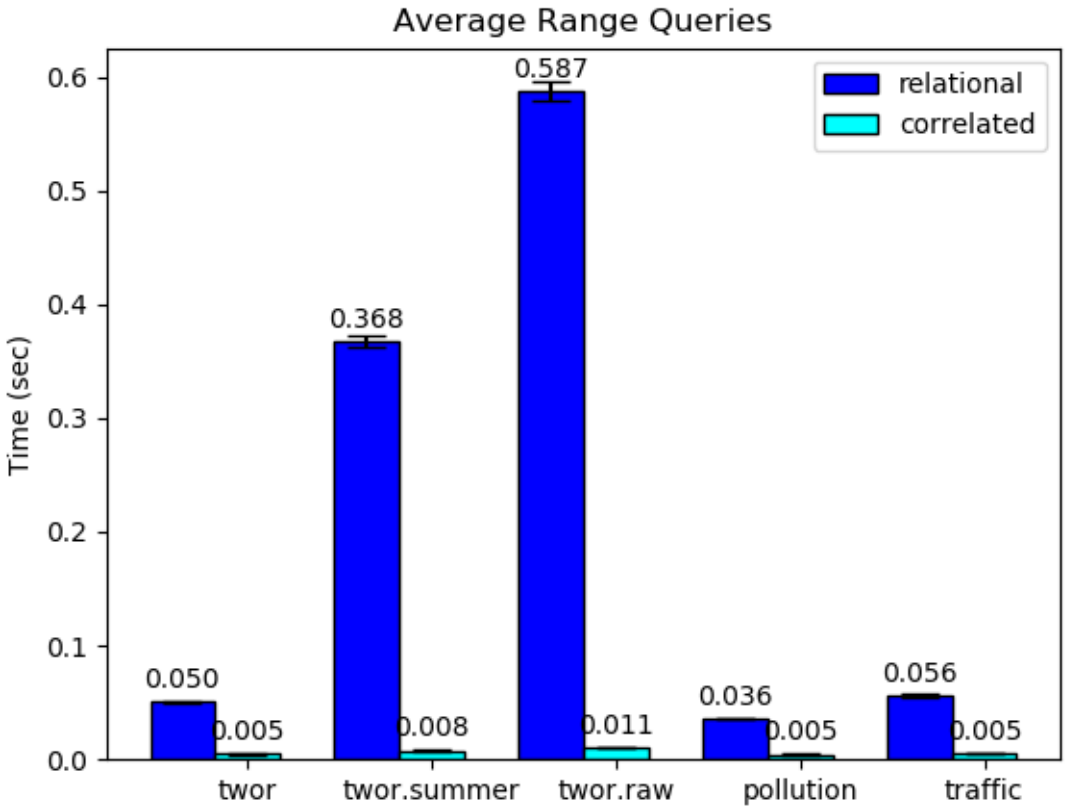
Και αυτή μια αρκετά βασική αναζήτηση σε χρονικά δεδομένα καθώς μπορεί να χρησιμεύσει στην διαμόρφωση μιας γενικής εικόνας για την λειτουργία ενός αισθητήρα για κάποιο χρονικό διάστημα. Π.χ. η μέση τιμή ενός αισθητήρα θερμοκρασίας μπορεί να μας δείξει διάφορα για την θέρμανση ή μη ενός δωματίου σπιτιού ή η μέση τιμή ενός επιταχυνσιόμετρου μπορεί να μας δείξει πολλά για τον τρόπο χρήσης ενός κινητού. Τα

SQL ερωτήματα που προτείνονται στην προκειμένη περίπτωση φαίνονται στον πίνακα 6.

Πίνακας 6 SQL Ερωτήματα μέσης τιμής σε δεδομένη χρονική περίοδο

Relational	*SELECT AVG(e.Value) FROM Events e WHERE e.sensor = :sensor AND e.Timestamp >= :time1 AND e.Timestamp <= :time2;*
Correlation	*SELECT AVG(v.value) FROM Events e, Sensor s, Value v WHERE e.sensor_id = s.id AND s.sensor_name = :sensor AND e.value_id = v.id AND e.Timestamp >= :time1 AND e.Timestamp <= :time2;*

όπου :sensor, :time1 και :time2 είναι οι παράμετροι για τον αισθητήρα και τα χρονικά όρια αντίστοιχα. Για παράδειγμα αν θεωρήσουμε ως :time1, :time2 τα 22-06-2015 22:32:00 και 22-06-2015 22:33:00 αντίστοιχα και ως :sensor τον M41 τότε το αποτέλεσμα θα είναι 0,375.



Σχήμα 3 Συγκριτικοί χρόνοι ερωτήματος μέσης τιμής σε συγκεκριμένη χρονική περίοδο

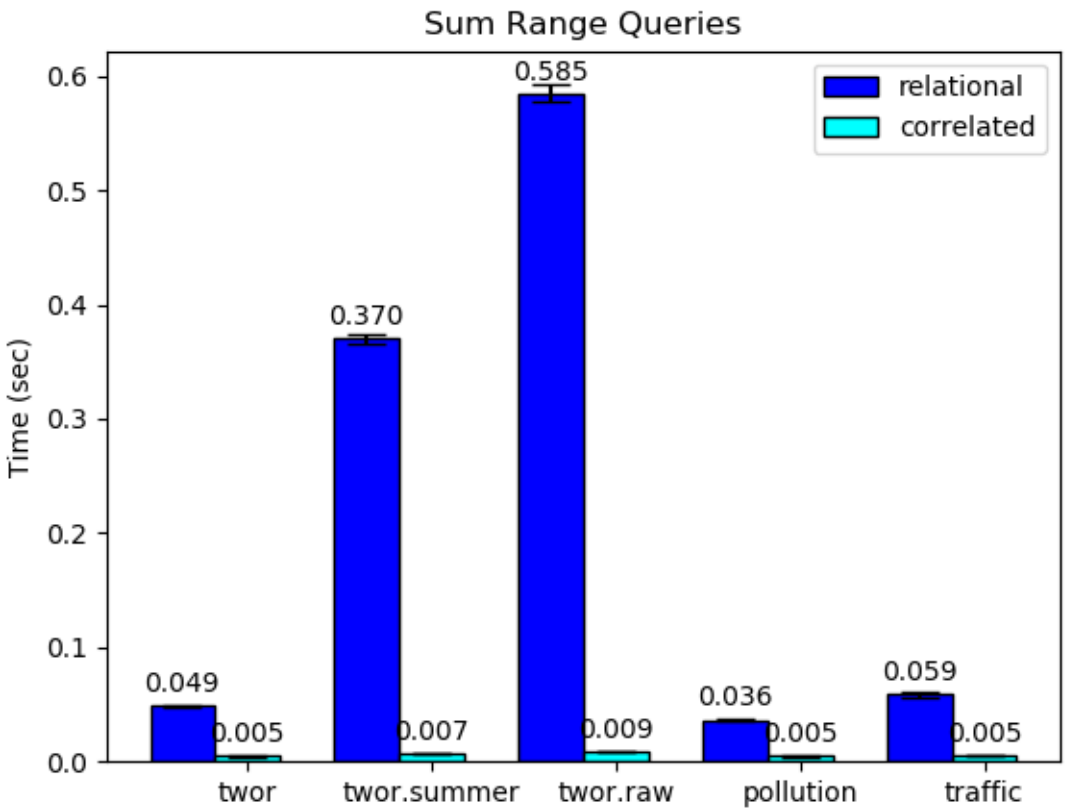
3.3 Άθροισμα τιμών ενός αισθητήρα σε μια δεδομένη χρονική περίοδο

Αρκετά παρόμοια περίπτωση με την προηγούμενη μόνο που αλλάζει η συναθροιστική συνάρτηση του αποτελέσματος. Το άθροισμα τιμών ενός αισθητήρα μπορεί να είναι χρήσιμο σε περιπτώσεις μέτρησης συνολικών βημάτων ή απόστασης. Τα SQL ερωτήματα που ενδείκνυνται στην προκειμένη περίπτωση φαίνονται στον πίνακα 7.

Πίνακας 7 SQL Ερωτήματα αθροίσματος τιμών σε συγκεκριμένη χρονική περίοδο

Relational	*SELECT SUM(e.Value) FROM Events e WHERE e.sensor = :sensor AND e.Timestamp >= :time1 AND e.Timestamp <= :time2;*
Correlation	*SELECT SUM(v.value) FROM Events e, Sensor s, Value v WHERE e.sensor_id = s.id AND s.sensor_name = :sensor AND e.value_id = v.id AND e.Timestamp >= :time1 AND e.Timestamp <= :time2;*

όπου :sensor, :time1 και :time2 είναι οι παράμετροι για τον αισθητήρα και τα χρονικά όρια αντίστοιχα. Για παράδειγμα αν θεωρήσουμε ως :time1, :time2 τα 22-06-2015 22:32:00 και 22-06-2015 22:33:00 αντίστοιχα και ως :sensor τον M41 τότε το αποτέλεσμα θα είναι 0,75.



Σχήμα 4 Συγκριτικοί χρόνοι ερωτήματος αθροίσματος σε συγκεκριμένη χρονική περίοδο

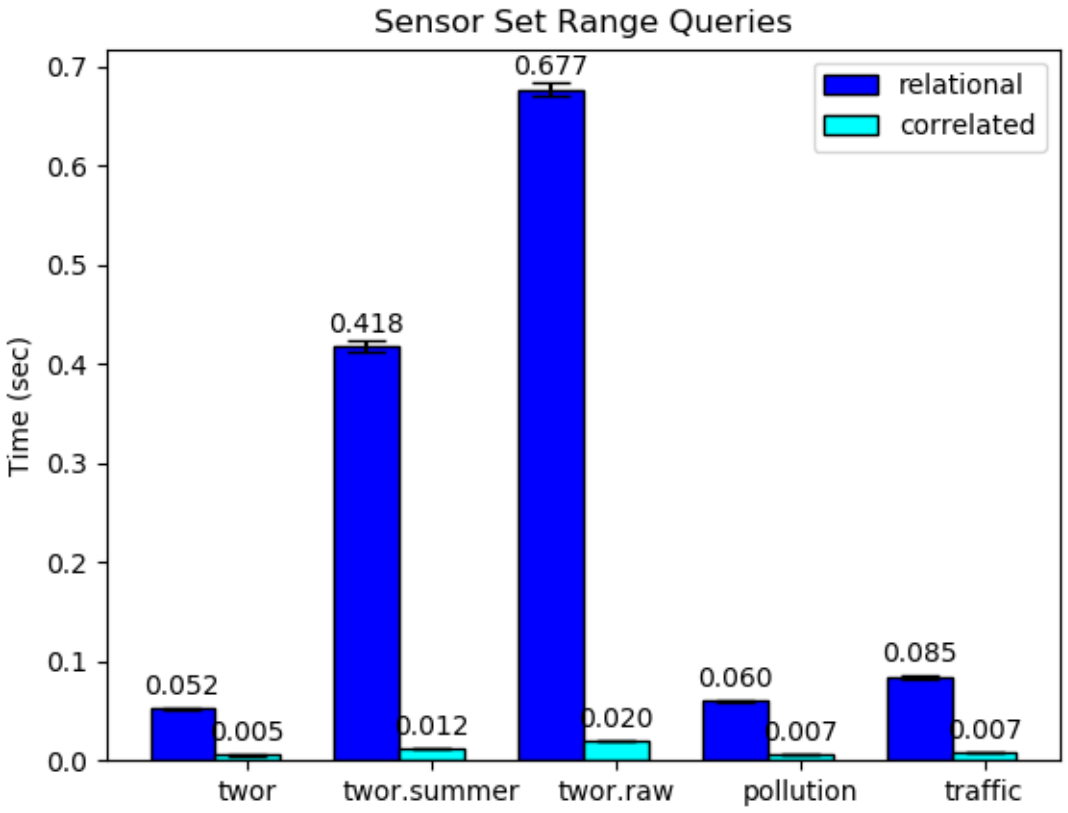
3.4 Μέση τιμή ενός συνόλου αισθητήρων σε μια δεδομένη χρονική περίοδο

Ίδια περίπτωση και αυτή με τις προηγούμενες με την διαφορά ότι τώρα μετράμε την συνολική μέση τιμή για πολλούς αισθητήρες μαζί. Αυτό μπορεί να είναι χρήσιμο σε περιπτώσεις που θέλουμε να συνδυάσουμε "κοντινούς" (λειτουργικά και χωρικά) μεταξύ τους αισθητήρες και θέλουμε να πάρουμε μια συνολική εικόνα για αυτούς. Τα SQL ερωτήματα που ενδείκνυνται στην προκειμένη περίπτωση φαίνονται στον πίνακα 8.

Πίνακας 8 SQL Ερωτήματα μέσης τιμής συνόλου σε δεδομένη χρονική περίοδο

Relational	*SELECT AVG(e.Value) FROM Events e WHERE e.sensor IN (:sensor_set) AND e.Timestamp >= :time1 AND e.Timestamp <= :time2;*
Correlation	*SELECT AVG(v.value) FROM Events e, Sensor s, Value v WHERE e.sensor_id = s.id AND s.sensor_name IN (:sensor_set) AND e.value_id = v.id AND e.Timestamp >= :time1 AND e.Timestamp <= :time2;*

όπου :sensor_set, :time1 και :time2 είναι οι παράμετροι για το σύνολο αισθητήρων και τα χρονικά όρια αντίστοιχα. Για παράδειγμα αν θεωρήσουμε ως :time1, :time2 τα 22-06-2015 22:32:00 και 22-06-2015 22:33:00 αντίστοιχα και ως :sensor_set το (T01, T02) τότε το αποτέλεσμα θα είναι 26.



Σχήμα 5 Συγκριτικοί χρόνοι ερωτήματος μέσης τιμής συνόλου σε δεδομένη χρονική περίοδο

Στα τρία τελευταία ερωτήματα παρατηρούμε ότι οι χρόνοι εκτέλεσης για το σχεσιακό σχήμα είναι ελάχιστα μεγαλύτεροι των αντίστοιχων χρόνων του πρώτου ερωτήματος ενώ οι χρόνοι του συσχετιστικού σχήματος κυμαίνονται στα ίδια επίπεδα. Αυτό μπορεί να συμβαίνει γιατί στο συσχετιστικό σχήμα έχουμε να κάνουμε με μία κατά στήλη αποθήκευση των τιμών πράγμα που ίσως κάνει την εφαρμογή μιας συνάρτησης συνάθροισης, όπως η μέση τιμή, πιο αποδοτική.

3.5 Εύρεση τιμής αισθητήρα x λεπτά μετά από δεδομένη χρονική στιγμή

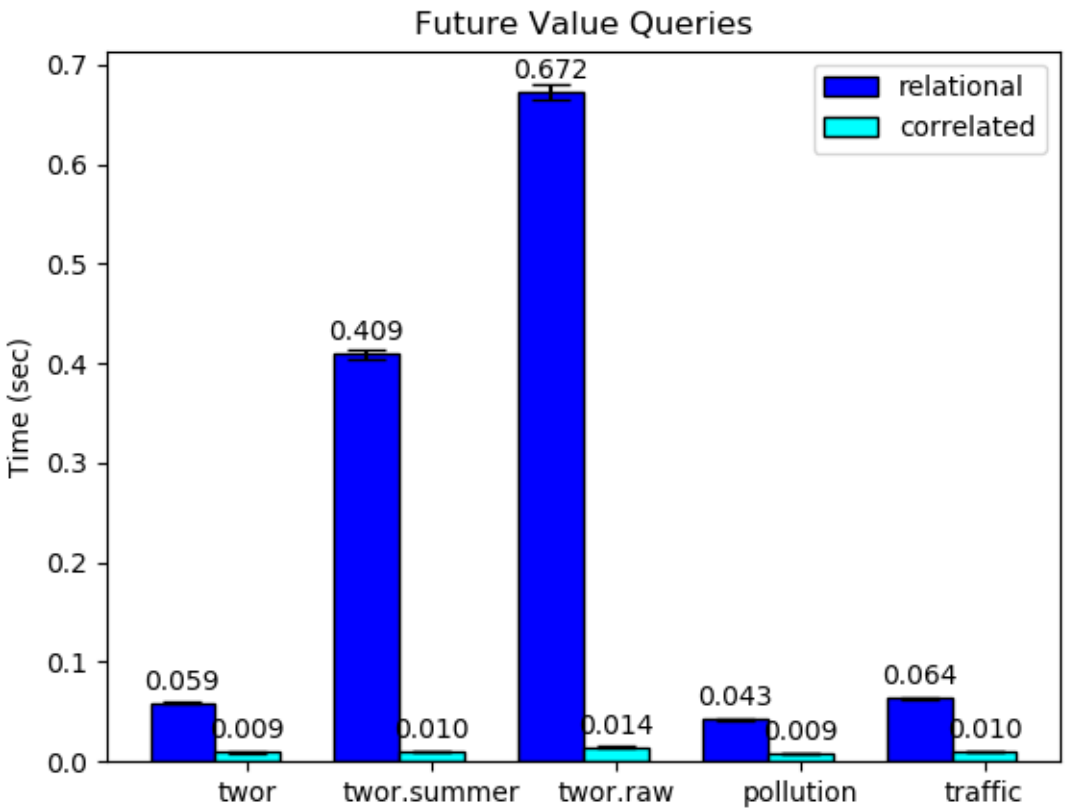
Πρόκειται για μία ελαφρώς διαφορετική περίπτωση από τις προηγούμενες αλλά αρκετά παρόμοια. Σε περιπτώσεις ζωντανών και on demand χρονικών δεδομένων θα χρειαζόταν ολόκληρο σύστημα κατηγοριοποίησης για να απαντηθεί. Στην περίπτωσή μας, όπου όλα τα δεδομένα είναι γνωστά εξ αρχής και αποθηκευμένα, το ερώτημα

γίνεται τετριμμένο. Η χρησιμότητα του βρίσκεται κυρίως στην εύρεση εξαρτήσεων ή συνθηκών μεταξύ των δεδομένων, δηλαδή αν ξέρουμε την τιμή για κάποια χρονική στιγμή να εξετάσουμε αν αυτό επηρεάζει την τιμή x λεπτά αργότερα κατά κάποιο τρόπο και αν αυτό είναι αναγκαία συνθήκη. Τα SQL ερωτήματα που ενδείκνυνται στην προκειμένη περίπτωση φαίνονται στον πίνακα 9.

Πίνακας 9 SQL Ερωτήματα τιμής αισθητήρα x λεπτά μετά από δεδομένη χρονική στιγμή

Relational	*SELECT e.Value FROM Events e WHERE e.Sensor = :sensor AND e.Timestamp <= DATE_ADD(:time, INTERVAL 10 MINUTE) ORDER BY e.Timestamp DESC LIMIT 1;*
Correlation	*SELECT v.value FROM Events e, Sensor s, Value v WHERE e.sensor_id = s.id AND s.sensor_name = :sensor AND e.value_id = v.id AND e.Timestamp <= DATE_ADD(:time, INTERVAL 10 MINUTE) ORDER BY e.Timestamp DESC LIMIT 1;*

όπου :sensor, :time είναι οι παράμετροι για τον αισθητήρα και την χρονική στιγμή αντίστοιχα. Για παράδειγμα αν θεωρήσουμε ως :time το 22-06-2015 22:22:10 και ως :sensor τον M41 τότε το αποτέλεσμα θα είναι 0,5.



Σχήμα 6 Συγκριτικοί χρόνοι ερωτήματος εύρεσης τιμής x λεπτά μετά από δεδομένη χρονική στιγμή

3.6 Εντοπισμός δυσλειτουργίας αισθητήρα

Μία αναζήτηση πιο ανεπτυγμένη από τις υπόλοιπες η οποία έχει περιθώρια εξερεύνησης και πειραματισμού. Σε δεδομένα πραγματικού χρόνου η αναζήτηση αυτή θα χρειαζόταν την ανάπτυξη ενός συστήματος κατηγοριοποίησης για την απάντησή της, όπως ένα νευρωνικό δίκτυο ή μια τεχνική κινούμενου παράθυρου (sliding window). Έχοντας όμως όλα τα δεδομένα αποθηκευμένα και γνωστά από την αρχή μπορούμε να σκεφτούμε μία μεθοδολογία αρκετά πιο απλή.

Αρχικά χρειάζεται να ορίσουμε πότε ένας αισθητήρας θεωρείται ότι είναι δυσλειτουργικός. Ποια είναι, δηλαδή, η παράμετρος που τον καθιστά ελαττωματικό. Θα θεωρήσουμε λοιπόν ότι ένας αισθητήρας δυσλειτουργεί όταν παρουσιάζει στις ενδείξεις του μεγάλο πλήθος τιμών που ξεφεύγουν κατά ένα ποσοστό από το μέσο όρο τιμών του, κοινώς όταν παρουσιάζει πολλά spikes. Φυσικά το ότι ένας αισθητήρας μπορεί να παρουσιάσει κάποιες ακραίες ενδείξεις δεν τον καθιστά αυτόματα δυσλειτουργικό. Πρέπει να βρεθεί το σωστό κατώφλι για το ποσοστό των ακραίων ενδείξεων που θεωρούμε ότι είναι ικανό να τον κατατάξει σε πιθανόν δυσλειτουργικό.

3.6.1 Sliding Window Fault Detection

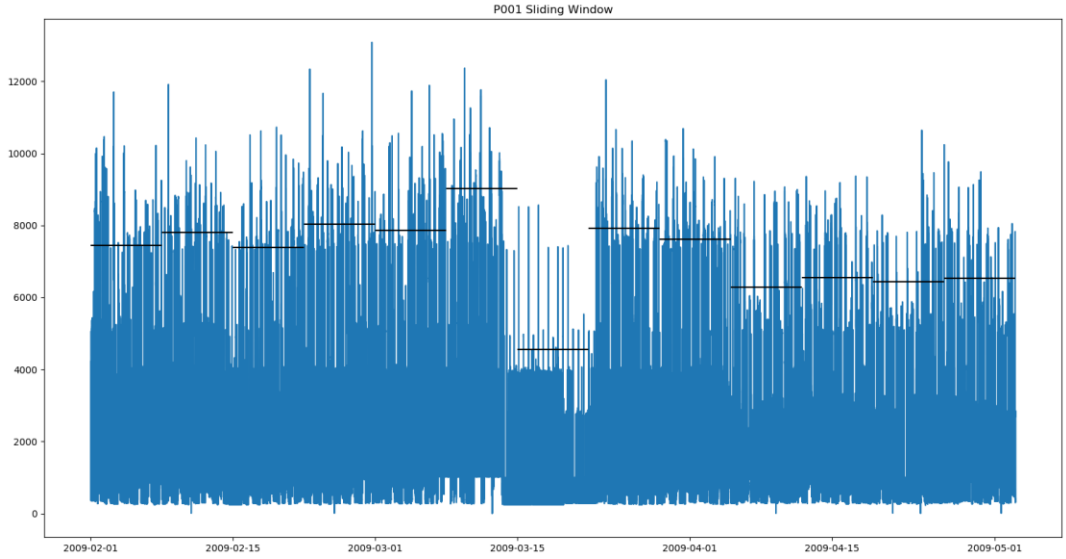
Η τεχνική κινούμενου παραθύρου (sliding window) είναι μία τεχνική ανάλυσης χρονικών σειρών. Με αυτή την τεχνική υποδιαιρούμε την σειρά σε χρονικά διαστήματα και την εξετάζουμε ξεχωριστά σε κάθε τέτοιο διάστημα. Στην περίπτωσή μας θα χωρίσουμε κάθε σειρά σε διαστήματα μιας εβδομάδας και θα αναλύσουμε τις μετρήσεις κάθε αισθητήρα μέσα σε αυτά. Η μεθοδολογία είναι η εξής:

- Υπολογισμός της μέσης τιμής (***m***) της χρονικής σειράς στο διάστημα
- Υπολογισμός της τυπικής απόκλισης (***σ***) της χρονικής σειράς στο διάστημα
- Απομόνωση των τιμών που είναι μεγαλύτερες κατ' απόλυτη τιμή από ***m + 2σ***
- Κατάταξη σε δυσλειτουργία αν το ποσοστό των τιμών αυτών είναι μεγαλύτερο του **10%** των συνολικών τιμών.
- Επανάληψη των παραπάνω ανά διάστημα 1 εβδομάδας δεδομένων.

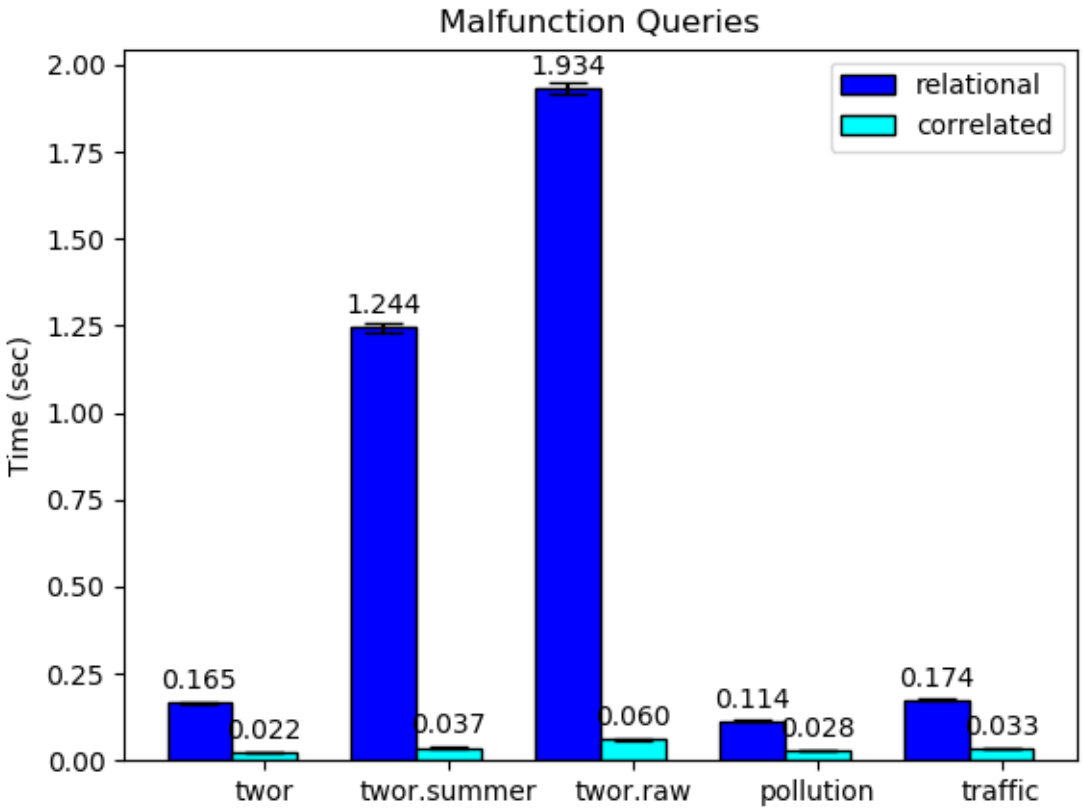
Μερικές παρατηρήσεις για την μεθοδολογία αυτή: Πρώτον, επιλέξαμε τη μέση τιμή γιατί είναι το πιο χαρακτηριστικό στατιστικό μέτρο για τον καθορισμό μιας ομαλής λειτουργίας του αισθητήρα. Δεύτερον επιλέξαμε την τυπική απόκλιση γιατί είναι ένα μέτρο που χρησιμοποιείται για να υπολογιστεί το ποσό της μεταβολής ή της διασποράς ενός συνόλου τιμών δεδομένων και εκφράζεται στις ίδιες μονάδες με τα δεδομένα, πράγμα σημαντικό για την εύρεση πιθανών ακραίων τιμών. Επίσης ο πολλαπλασιαστής 2 για την τυπική απόκλιση επιλέχθηκε αφηρημένα βάσει λογικής. Τέλος το ποσοστό-κατώφλι που επιλέχθηκε για την κατάταξη δυσλειτουργίας προέκυψε μέσα από πειραματισμό και είναι σχετικά υψηλό για να αποφευχθούν πολλές περιπτώσεις από false-positives.

Εφαρμόσαμε τη μεθοδολογία αυτή σε όλους τους αριθμητικούς αισθητήρες από κάθε dataset. Τα αποτελέσματα έδειξαν μόνο έναν υποψήφιο αισθητήρα. Πρόκειται για έναν αισθητήρα μέτρησης κατανάλωσης ρεύματος ("P001") στο καλοκαιρινό dataset. Πράγματι παρατηρώντας στο σχήμα 7 το διάγραμμα τιμών μπορεί να πει κάποιος ότι ο αισθητήρας παρουσιάζει κάποια πιθανή δυσλειτουργία με τις οριζόντιες γραμμές να δείχνουν την τιμή m+2σ για κάθε χρονικό παράθυρο. Οι συνθήκες όμως που επικρατούσαν κατά την διεξαγωγή του πειράματος (καλοκαιρινοί μήνες, εκτεταμένη χρήση κλιματιστικού σε ζεστές μέρες πράγμα που μεγαλώνει σημαντικά την κατανάλωση ρεύματος σε σχέση με τον μέσο όρο κλπ.) μπορούν να δικαιολογήσουν κάπως την συμπεριφορά του αισθητήρα και να τον κατατάξουν σε πιθανό false positive κρούσμα. Όμως με βάση τη μεθοδολογία αυτή και με κανονικές συνθήκες αν ένας

αισθητήρας παρουσιάσει παρόμοια συμπεριφορά τότε πιάνεται στις πιθανές περιπτώσεις ελαττωματικών ενδείξεων.



Σχήμα 7 Διάγραμμα τιμών αισθητήρα P001.



Σχήμα 8 Συγκριτικοί χρόνοι ερωτημάτων δυσλειτουργίας

3.6.2 ARIMA Forecast

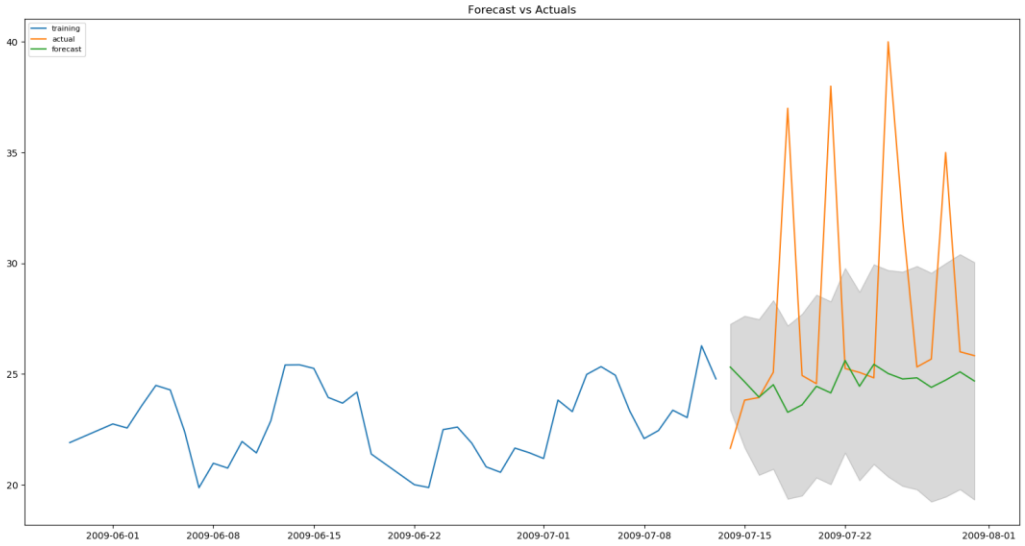
Μια άλλη τεχνική, γνωστή για την χρήση της στην μελέτη χρονικών σειρών, είναι το μοντέλο ARIMA (AutoRegressive Integrated Moving Average). Το μοντέλο αυτό είναι μια γενίκευση του μοντέλου ARMA και χρησιμοποιείται στις χρονικές σειρές για την μελέτη τους ή για την πρόβλεψη μελλοντικών τους σημείων. Μπορεί να χρησιμοποιηθεί επίσης και για την ανίχνευση ανωμαλιών στα δεδομένα που είναι και η περίπτωσή μας. Σε αυτή την παράγραφο προτείνουμε μια ανάλυση όπου χρησιμοποιώντας το μοντέλο ARIMA θα προβλέψουμε μελλοντικές τιμές των αισθητήρων και μέσω αυτών θα εντοπίσουμε περιπτώσεις δυσλειτουργίας.[11]

Στο Autoregressive μέρος του μοντέλου (συμβ. AR(p)) η μελλοντική τιμή της μεταβλητής θεωρείται ότι είναι ένας γραμμικός συνδυασμός μιας παλιάς τιμής της μεταβλητής, μιας σταθεράς και λευκού θορύβου.

Στο Moving average μέρος του μοντέλου (συμβ. MA(q)) οι παλιές τιμές της μεταβλητής χρησιμοποιούνται ως επεξηγηματικές μεταβλητές. [12]

Τέλος το Integrated κομμάτι του μοντέλου αναφέρεται στην απαλοιφή της στατικότητας της σειράς με την εφαρμογή πεπερασμένης διαφοράς μεταξύ των σημείων της σειράς.

Αρχικά θα μελετήσουμε μέσω του μοντέλου ARIMA την συμπεριφορά των συνεχών αισθητήρων, όπως οι αισθητήρες θερμοκρασίας και ισχύος. Για κάθε αισθητήρα θα χρησιμοποιήσουμε το 70% των μετρήσεων ως training set και το 30% ως testing. Επίσης θα προσθέσουμε τεχνητά και μερικά "ελαττωματικά" σημεία στο testing set μας για να εξετάσουμε την αξιοπιστία του μοντέλου. Τα αποτελέσματα φαίνονται στο Σχήμα 9. Βλέπουμε ότι για έναν αισθητήρα θερμοκρασίας οι ακραίες τιμές που προσθέσαμε τεχνητά βρίσκονται εκτός του 95% confidence interval (γκρι περιοχή) της πρόβλεψής μας πράγμα που μας προσφέρει μια ένδειξη της αξιοπιστίας του μοντέλου μας.



Σχήμα 9 Πρόβλεψη αισθητήρα θερμοκρασίας με το μοντέλο ARIMA

Για τη μαθηματική εξακρίβωση της αξιοπιστίας του μοντέλου χρησιμοποιήσαμε τρεις μετρικές για κάθε dataset. Το μέτρο MAPE (Mean Absolute Percentage Error), Correlation και MinMax Error. Ο λόγος που χρησιμοποιούμε αυτές τις μετρικές είναι γιατί πρόκειται για ποσοστιαίες μετρικές και ως εκ τούτου είναι ανεξάρτητες του μεγέθους των δεδομένων. Τα αποτελέσματα φαίνονται στον Πίνακα 10.

Πίνακας 10 Μετρικές στατιστικής επίδοσης

	MAPE	Correlation	MinMax
CASAS	0.07	0.46	0.065
Pollution	0.05	0.73	0.04
Traffic	0.09	0.6	0.082

Με μέγιστο ποσοστό MAPE το 9% θεωρούμε ότι το μοντέλο μας κάνει αρκετά καλή δουλειά στην ανίχνευση πιθανών μελλοντικών ανωμαλιών με μικρή πιθανότητα για ψευδώς θετικές ενδείξεις.

Το δεύτερο μέρος χρήσης του μοντέλου έχει να κάνει με τη συσχέτιση αισθητήρων ανάλογα με την φυσική τους απόσταση στο χώρο. Σε αυτά τα σετ αισθητήρων οι τιμές του καθενός ξεχωριστά ενώ μας λένε κάποια πράγματα σχετικά με την ορθή λειτουργία του, εξετάζοντας τις τιμές τους σε σύνολα μπορούμε να εξάγουμε ακόμα πιο σημαντικές

πληροφορίες. Για παράδειγμα αν έχουμε 2 αισθητήρες θερμοκρασίας σε κοντινά δωμάτια θα περιμένουμε οι ενδείξεις τους σε μια χρονική στιγμή να μην έχουν μεγάλη απόκλιση ή αν έχουμε 2 κοντινούς αισθητήρες κίνησης θα περιμένουμε ότι αν ενεργοποιηθεί ο ένας τότε κατά μεγάλη πιθανότητα θα ενεργοποιηθεί και ο άλλος αμέσως μετά. Για τον σωστό καταρτισμό των συνόλων θα πρέπει: α) να καθορίσουμε ένα καλό μέτρο για το πότε ένα ζευγάρι αισθητήρων θεωρείται ότι αλληλοεπηρεάζεται και β) να ευθυγραμμίσουμε τις τιμές του εξεταζόμενου αισθητήρα με τις τιμές του συνόλου στο οποίο ανήκει και να "εκπαιδεύσουμε" το μοντέλο με βάση τις τιμές όλου του συνόλου.

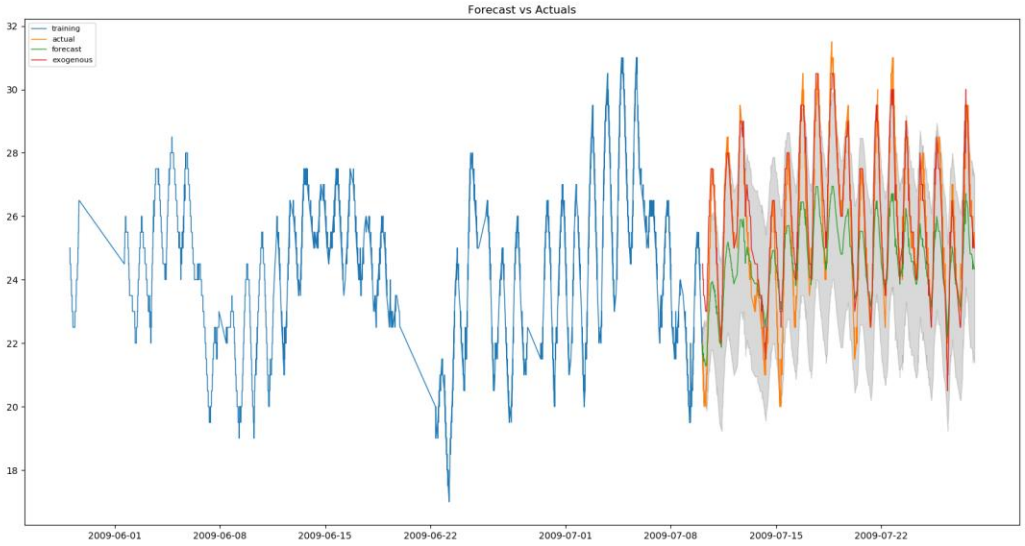
Η απάντηση στο πρώτο ερώτημα μπορεί να δοθεί με διάφορους τρόπους. Η εξάρτηση δύο αισθητήρων μπορεί να καθορίζεται είτε από τη θέση τους στο δωμάτιο, είτε από κάποιο αντικείμενο που είναι συνδεδεμένοι είτε από τον χρήστη που πυροδότησε τον αισθητήρα σε μια συγκεκριμένη στιγμή. Στην μελέτη μας θα χρησιμοποιήσουμε την φυσική θέση του αισθητήρα καθώς αυτό είναι το πιο ξεκάθαρο μέτρο σύγκρισης που μας δίνεται. Συγκεκριμένα θα χρησιμοποιήσουμε μία μετρική πιο ισχυρού σήματος, δηλαδή όσο πιο κοντά είναι ένας γειτονικός αισθητήρας στον στόχο τότε έχει μεγαλύτερο βάρος στο μοντέλο.

Για το δεύτερο ερώτημα ευτυχώς το μοντέλο ARIMA μας προσφέρει μια παραλλαγή στην οποία μπορούμε να εισάγουμε και μία εξωγενή μεταβλητή στο μοντέλο και να προβλέψουμε με βάση αυτήν.

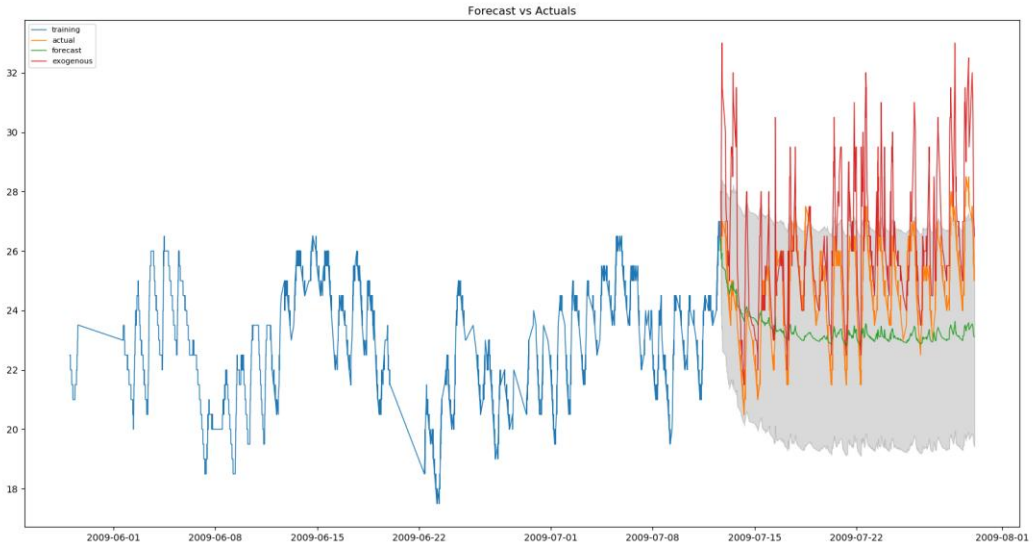
Με βάση τα παραπάνω επιλέγουμε δύο αισθητήρες κοντινούς μεταξύ τους από το CASAS dataset και παρουσιάζουμε το αποτέλεσμα στα σχήματα 10 και 11.

Στο σχήμα 10 συγκρίνουμε δύο αισθητήρες θερμοκρασίας σε 2 διπλανές κρεβατοκάμαρες. Παρατηρούμε ότι λίγο πολύ οι μετρήσεις είναι σε συμφωνία μεταξύ τους και το μοντέλο μας προβλέπει μια ομαλή ροή μετρήσεων.

Στο σχήμα 11 παρότι έχουμε δύο κοντινούς αισθητήρες, παρατηρούμε μια σχετική απόκλιση στις μετρήσεις του μοντέλου μας. Αυτό μπορεί να εξηγείται από το γεγονός ότι ο αισθητήρας που εξετάζουμε βρίσκεται στο σαλόνι ενώ ο άλλος στην κουζίνα που ως περιβάλλον αναμένουμε να έχει κάπως πιο υψηλές ενδείξεις σε κάποια διαστήματα της ημέρας.



Σχήμα 10 Διάγραμμα πρόβλεψης με εξωγενή μεταβλητή



Σχήμα 11 Διάγραμμα πρόβλεψης με εξωγενή μεταβλητή με αποκλίσεις

Για να αποφύγουμε αυτήν την διακύμανση μπορούμε να προσθέσουμε μία ακόμα ανεξάρτητη χρήση του μοντέλου ARIMA πριν από την χρήση του με εξωγενή μεταβλητή. Πιο συγκεκριμένα θα μπορούσαμε να εφαρμόσουμε το μοντέλο χωρίς εξωγενή παράγοντα και να αξιολογήσουμε τη διαφορά της πρόβλεψης για κάθε σημείο με την αληθινή τιμή κάθε σημείου και αν αυτή είναι πάνω από ένα κατώφλι να θεωρήσουμε την ένδειξη σε αυτό το χρονικό σημείο ως πιθανώς λανθασμένη. Αν η ένδειξη αυτή εμφανιστεί και στο επόμενο βήμα ως πιθανώς λανθασμένη τότε την κατατάσσουμε στις λανθασμένες ενδείξεις.

Το κατώφλι μπορεί να οριστεί ως η τιμή $|\mu + 3\sigma|$ όπου μ είναι η μέση τιμή της χρονικής σειράς και σ η τυπική απόκλιση. [13]

3.7 Εύρεση της δραστηριότητας του χρήστη δεδομένης της τιμής ενός αισθητήρα

Ένα ερώτημα με το οποίο μπορούμε να συσχετίσουμε την τιμή ενός αισθητήρα με κάποια δραστηριότητα που κάνει ο χρήστης. Μέσω αυτού μπορούμε να βγάλουμε χρήσιμα συμπεράσματα για τη συμπεριφορά ενός αισθητήρα βάζοντας ένα πλαίσιο λειτουργίας γύρω από τις τιμές του. Για παράδειγμα μπορούμε να περιμένουμε ότι όταν ένας αισθητήρας ζεστού νερού έχει υψηλή τιμή τότε είναι πολύ πιθανό ο χρήστης να κάνει μπάνιο ή να πλένει τα πιάτα. Το πρόβλημα με αυτό το ερώτημα είναι ότι χρειάζεται κάποιο annotated dataset για να απαντηθεί, το οποίο είναι αρκετά περιορισμένου εύρους σε σχέση με τα raw δεδομένα και πολλές φορές είναι συμπληρωμένο χειροκίνητα με ότι κακοτεχνίες μπορεί να συνεπάγεται αυτό. Τα SQL ερωτήματα για αυτήν την κατηγορία φαίνονται στον πίνακα 11.

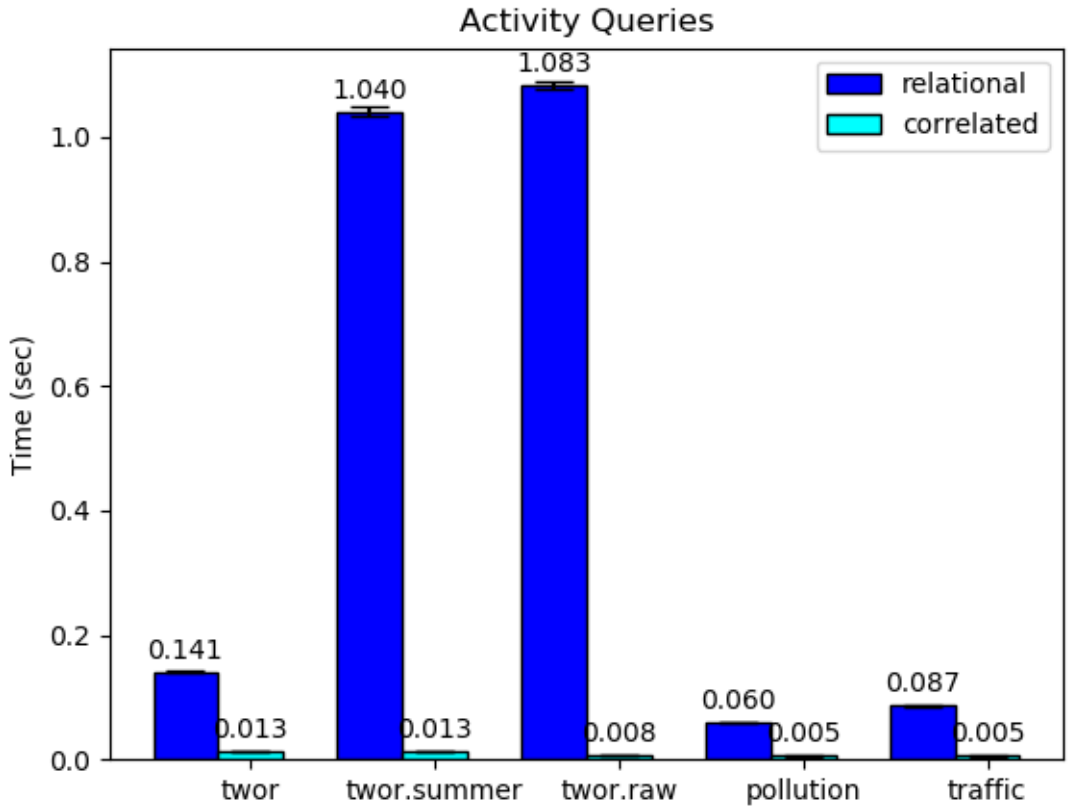
Πίνακας 11 SQL Ερωτήματα προσδιορισμού δραστηριότητας βάσει κατάστασης

Relational	*SELECT e.Activity FROM Events e WHERE e.Timestamp = (SELECT MAX(e1.Timestamp) FROM Events e1 WHERE e1.Activity is not null AND e1.Timestamp <= (SELECT MAX(e2.Timestamp) FROM Events e2 WHERE e2.Sensor = :sensor AND e2.Value = :value AND e2.Timestamp <= :time));*
Correlation	*SELECT a.activity_name FROM Events e, Activity a WHERE e.activity_id = a.id AND e.Timestamp = (SELECT MAX(e1.Timestamp) FROM Events e1 WHERE e1.activity_id is not null AND e1.Timestamp <= (SELECT MAX(e2.Timestamp) FROM Events e2, Sensor s, Value v WHERE e2.sensor_id = s.id AND s.sensor_name = :sensor AND e2.value_id = v.id AND v.value = :value AND e2.Timestamp <= :time));*

όπου :sensor είναι η παράμετρος για τον αισθητήρα, :value η παράμετρος για την τιμή του αισθητήρα και :time η παράμετρος για την χρονική στιγμή. Για παράδειγμα αν θεωρήσουμε ως :time το 22-06-2015 22:33:00, ως :sensor τον T01 και ως :value το 25 τότε το αποτέλεσμα θα είναι Activity1.

Στα συγκεκριμένα ερωτήματα ψάχνουμε να βρούμε την δραστηριότητα που κάνει ένας χρήστης με βάση τη κατάσταση ενός αισθητήρα. Έτσι στο εσωτερικό Select ερώτημα

παίρνουμε την χρονική στιγμή ενός γεγονότος προσδιορίζοντας την κατάσταση του αισθητήρα μέσω της τιμής του, του ονόματός του και μια χρονική στιγμή. Στο ενδιάμεσο Select ερώτημα παίρνουμε το πιο πρόσφατα καταγεγραμμένο γεγονός στην κατάσταση του αισθητήρα που ορίσαμε, στο οποίο προσδιορίζεται το πεδίο της δραστηριότητας. Τέλος στο εξωτερικό Select ερώτημα παίρνουμε την ετικέτα της δραστηριότητας. Στο σχήμα 11 παρατηρούμε ότι δεν υφίστανται χρόνοι για το τελευταίο dataset διότι πρόκειται για μη annotated dataset πράγμα που αναδεικνύει και την αδυναμία εφαρμογής αυτού του ερωτήματος σε κάθε περίπτωση.



Σχήμα 12 Συγκριτικοί χρόνοι ερωτήματος εύρεσης δραστηριότητας

3.8 Σύγκριση μεγέθους μεταξύ σχεσιακής και συσχετιστικής ΒΔ

Πρόκειται για το τελικό ερώτημα και τον αντικειμενικό σκοπό της μελέτης μας. Εξετάζουμε τελικά για κάθε ένα σύνολο δεδομένων πόσο χώρο κερδίζουμε χρησιμοποιώντας την συσχετιστική τεχνική αποθήκευσης σε σχέση με την σχεσιακή. Όπως αναφέραμε προηγουμένως, επειδή δεν βρήκαμε κάποιο ανοιχτού κώδικα συσχετιστικό ΣΔΒΔ, το περιβάλλον στο οποίο εξετάζουμε την συσχετιστική τεχνική είναι κατά στήλη αποθήκευση. Για αυτό τον λόγο θα χρησιμοποιήσουμε για την μέτρηση το μέγεθος των καθαρών δεδομένων και όχι όλης της βάσης όπως αποθηκεύεται στο

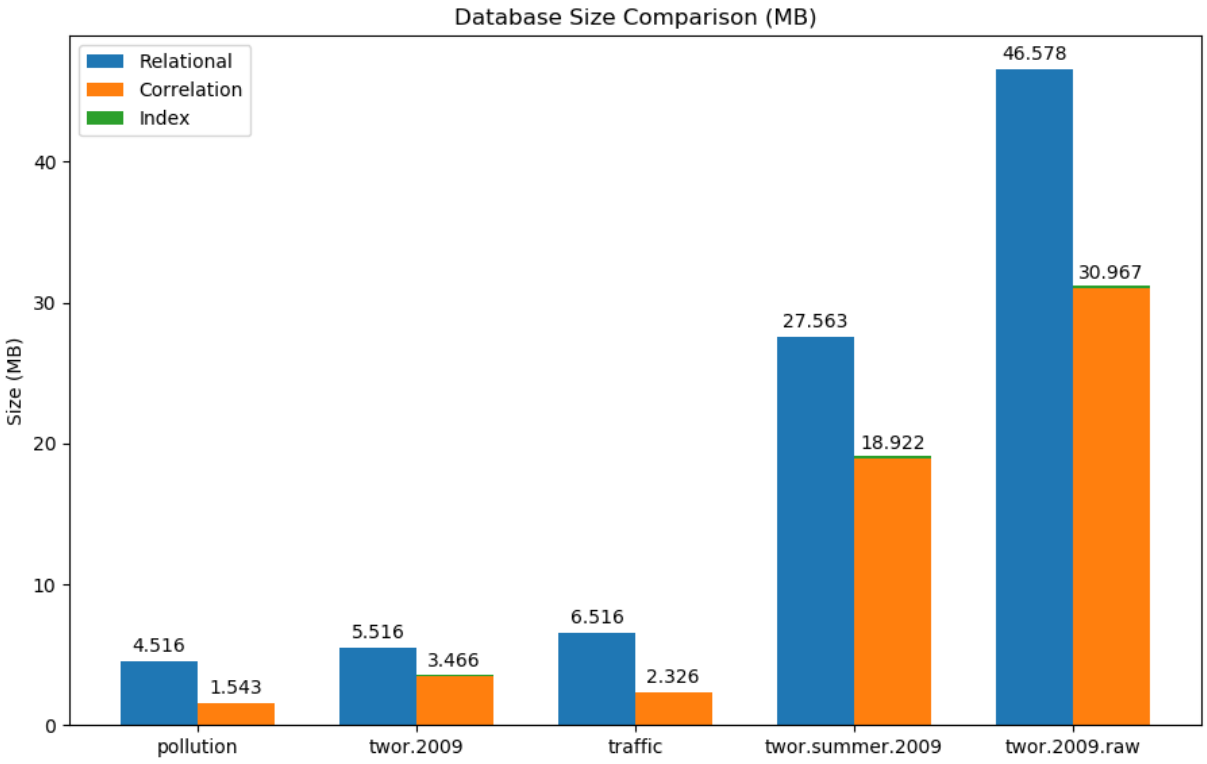
δίσκο γιατί μπορεί να υπάρχουν και έξτρα ευρετήρια ή μεταδεδομένα που σε ένα εγγενώς συσχετιστικό ΣΔΒΔ δεν θα υπήρχαν βάσει ορισμού.

Το ερώτημα που χρησιμοποιήσαμε είναι το εξής:

SELECT table_schema, ROUND(SUM(data_length) / 1024 / 1024, 1) FROM information_schema.tables GROUP BY table_schema;

Το ερώτημα αυτό υπολογίζει σε MB το συνολικό χώρο που καταλαμβάνουν τα δεδομένα κάθε σχήματος μέσα στον MySQL server μας.

Τα αποτελέσματα είναι τα παρακάτω:



Σχήμα 13 Μεγέθη Σχεσιακής και Συσχετιστικής ΒΔ

Στο σχήμα 13 παρατηρούμε ένα μοτίβο συμπεριφοράς που επιβεβαιώνει όσα ειπώθηκαν νωρίτερα για την συνεισφορά που μπορεί να έχει ένα συσχετιστικό σχήμα. Σε πιο μικρά μεγέθη δεδομένων η διαφορά μεγέθους είναι σχετικά μικρή σε σχέση με το σχεσιακό σχήμα. Αντίθετα όσο μεγαλώνει το μέγεθος των δεδομένων τόσο φαίνεται το κέρδος χώρου που μπορεί να μας παρέχει ένα συσχετιστικό σχήμα δεδομένων. Αυτό συμβαίνει διότι όσο αυξάνεται το πλήθος των δεδομένων μας τόσο αυξάνεται και η επανάληψη πληροφορίας και η περιοδικότητα μέσα σε αυτά αφήνοντας περιθώριο για μεγαλύτερη συμπίεση τιμών χωρίς απώλεια πληροφορίας.

4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή την εργασία μελετήσαμε μία διαφορετική τεχνική αναπαράστασης χρονικών δεδομένων για την εξοικονόμηση χώρου και εύρους δικτύου κατά την μεταφορά και την αποθήκευσή τους αντίστοιχα. Υιοθετήσαμε μια νέα τεχνική αναπαράστασης, που ονομάζεται Συσχετιστική ΒΔ, και επιτρέπει την εξάλειψη της επανάληψης πληροφορίας μέσω ενός ενιαίου λεξικού τιμών. Σε αντίθεση με τα παραδοσιακά σχεσιακά μοντέλα ή ακόμη και τα πιο πρόσφατα NoSQL μοντέλα που αποθηκεύουν όπως είναι κάθε χρονικό συμβάν, η συσχετιστική αναπαράσταση τα αναλύει σε μοναδικές τιμές και αποθηκεύει έναν δείκτη στην κεντρική δομή.

Αρχικά παρουσιάσαμε ένα σχεσιακό λογικό σχήμα για την αναπαράσταση των δεδομένων μας. Είδαμε ότι, παρότι αποτελεί την πιο κλασική και δημοφιλή μέθοδο αποθήκευσης, έχει κάποιους περιορισμούς που υπονομεύουν την χρησιμότητά της, όπως η ύπαρξη προκαθορισμένου σχήματος και η ανάγκη διαβάσματος ολόκληρης εγγραφής ως μονάδα.

Έπειτα παρουσιάσαμε τα χαρακτηριστικά μιας συσχετιστικής βάσης δεδομένων. Αυτό που χαρακτηρίζει μια τέτοια ΒΔ είναι ότι, αντί να αποθηκεύει τις πλειάδες όπως είναι, κρατάει ένα λεξικό ή ευρετήριο με όλες τις μοναδικές τιμές που εμφανίζονται στη βάση και χρησιμοποιεί "δείκτες" σε αυτό για την σύνθεση των αρχικών πλειάδων.

Ακόμη είδαμε τα βασικά χαρακτηριστικά των datasets που χρησιμοποιήσαμε, όπως το μέγεθος τους, το χρονικό διάστημα που καλύπτουν και το πλήθος των αισθητήρων που περιέχουν. Επιλέξαμε όλα τα datasets να έχουν ένα ικανό πλήθος αισθητήρων και διάστημα κάλυψης έτσι ώστε να έχουμε ένα αξιόπιστο δείγμα για τα μετέπειτα πειράματα.

Τέλος προτείναμε κάποια ενδιαφέροντα ερωτήματα για ανάλυση πάνω στα δύο διαφορετικά σχήματα που εξετάζουμε. Τα ερωτήματα αυτά περιλαμβάνουν από απλές περιπτώσεις, όπως η ανάκτηση των τιμών αισθητήρων για κάποιο χρονικό διάστημα, μέχρι πιο σύνθετες, όπως η εύρεση της δραστηριότητας που κάνει ένα άτομο με βάση την κατάσταση του αισθητήρα ή η εύρεση δυσλειτουργιών στους αισθητήρες. Μέσω των ερωτημάτων αυτών αναδείξαμε τα βασικά πλεονεκτήματα και οφέλη του συσχετιστικού σχήματος και βρήκαμε μια πιο συμπαγή μέθοδο αποθήκευσης των δεδομένων. Πρώτον είδαμε μέσω διαγραμμάτων ότι, παρά την ύπαρξη ενός ενδιάμεσου επιπέδου για την ανάκτηση των πραγματικών τιμών (λεξικό), το συσχετιστικό σχήμα είναι στις περισσότερες περιπτώσεις πιο γρήγορο στην απάντηση των ερωτημάτων από το σχεσιακό. Δεύτερον παρατηρήσαμε ότι όσο το μέγεθος των δεδομένων αυξάνει και η

επαναλαμβανόμενη πληροφορία γίνεται περισσότερη τόσο κερδίζει σε αποθηκευτικό χώρο - άρα και εύρος δικτύου - το συσχετιστικό σχήμα έναντι του σχεσιακού.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Correlation	Συσχετιστικός
Database	Βάση Δεδομένων
Queries	Ερωτήματα
Dataset	Σύνολο Δεδομένων
Annotation	Υποσημείωση
Sliding window	Κινούμενο παράθυρο
Confidence Interval	Διάστημα Εμπιστοσύνης

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

ΣΔΒΔ	Σύστημα Διαχείρισης Βάσεων Δεδομένων
VBS	Value Based Storage
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
WSU	Washington State University
CASAS	Center for Advanced Studies in Adaptive Systems
ARIMA	AutoRegressive Integrated Moving Average

ΑΝΑΦΟΡΕΣ

[1] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas, Collecting and disseminating smart home sensor data in the CASAS project, Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, 2009.

[2] D. Cook and M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, Methods of Information in Medicine, 2009.

[3] Chen, Feng & Deng, Pan & Wan, Jiafu & Zhang, Daqiang & Vasilakos, Athanasios & Rong, Xiaohui. (2015). Data Mining for the Internet of Things: Literature Review and Challenges. International Journal of Distributed Sensor Networks. 2015. pp. 1-14.

[4] Johnston, Megan & Campbell, Craig & Hayward, Rachel & Lowerison, Mark & Noonan, Vanessa & Pfister, Ted & Maxwell, Colleen & Fortin, Claire & Smith, Eric & Mah, Jean & Kapral, Moira & Jette, Nathalie & Pringsheim, Tamara & Korngut, Lawrence. (2013). Registry Data Storage and Curation. The Canadian journal of neurological sciences. pp. 4-5

[5] Wikipedia contributors, 'Correlation database', Wikipedia, The Free Encyclopedia, 13 August 2017; https://en.wikipedia.org/w/index.php?title=Correlation_database&oldid=795299401 [Προσπελάστηκε 15/10/19].

[6] Ashley, 'Time series analysis with pandas', Coding Club, 7 January 2019; https://ourcodingclub.github.io/2019/01/07/pandas-time-series.html [Προσπελάστηκε 23/10/19].

[7] Jennifer Walker, 'Tutorial: Time Series Analysis with Pandas', Dataquest, 10 January 2019; https://www.dataquest.io/blog/tutorial-time-series-analysis-with-pandas/ [Προσπελάστηκε 5/11/19].

[8] Stefan Bischof, Athanasios Karapantelakis, Cosmin-Septimiu Nechifor, Amit Sheth, Alessandra Mileo and Payam Barnaghi, Semantic Modeling of Smart City Data, Position Paper in W3C Workshop on the Web of Things: Enablers and services for an open Web of Devices, 25-26 June 2014, Berlin, Germany.

[9] Muhammad Intizar Ali, Feng Gao and Alessandra Mileo, CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets, The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, October 11-15, 2015, Bethlehem, PA, USA.

[10] R. Tönjes, P. Barnaghi, M. Ali, A. Mileo, M. Hauswirth, F. Ganz, S. Ganea, B. Kjærgaard, D. Kuemper, S. Nechifor, D. Puiu, A. Sheth, V. Tsiatsis, L. Vestergaard, Real Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications, poster session, European Conference on Networks and Communications 2014.

[11] Wikipedia contributors, 'Autoregressive integrated moving average', Wikipedia, The Free Encyclopedia, 13 April 2020, 01:39 UTC, <https://en.wikipedia.org/w/index.php?title=Autoregressive_integrated_moving_average&oldid=950625193> [Προσπελάστηκε 1/4/20].

[12] R. Adhikari and R. Agrawal, "An introductory study on time series modeling and forecasting," arXiv preprint arXiv:1302.6613, 2013.

[13] Yu, Tianqi & Akhtar, Auon & Wang, Xianbin & Shami, Abdallah, Temporal and spatial correlation based distributed fault detection in wireless sensor networks, Canadian Conference on Electrical and Computer Engineering 2015.

[14] Joseph Foley, Comparison of Data Warehousing DBMS Platforms, illuminate Solutions, 2009.

[15] SAS Institute Inc, Internet of Things What is it and why it matters, 2018; https://www.sas.com/el_gr/insights/big-data/internet-of-things.html#industries [Προσπελάστηκε 20/06/20].

[16] James E. Powell, Illuminate's Correlation Database Accelerates, Expands BI Queries, 4 September 2008; https://tdwi.org/articles/2008/04/09/illuminates-correlation-database-accelerates-expands-bi-queries.aspx [Προσπελάστηκε 20/06/2020].