ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανασκόπηση Τεχνικών Ομαδοποίησης για την Εξόρυξη Δεδομένων από το Διαδίκτυο των Πραγμάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλόη – Αφροδίτη Αντωνίου

Επιβλέπων: Διονύσιος-Δημήτριος Κουτσούρης
Καθηγητής ΕΜΠ

Αθήνα, Μάρτιος 2016

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Ανασκόπηση Τεχνικών Ομαδοποίησης για την Εξόρυξη Δεδομένων από το Διαδίκτυο των Πραγμάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλόη – Αφροδίτη Αντωνίου

Επιβλέπων: Διονύσιος-Δημήτριος Κουτσούρης
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Μαρτίου 2016

.....	.....	.....
Δ.Δ. Κουτσούρης	Κ. Νικήτα	Γ. Ματσόπουλος
Καθηγητής ΕΜΠ	Καθηγήτρια ΕΜΠ	Αν. Καθηγητής ΕΜΠ

Αθήνα, Μάρτιος 2016

.....

Αλόη – Αφροδίτη Αντωνίου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

Περίληψη

Το Διαδίκτυο των Πραγμάτων (ΔτΠ), ως αναδυόμενη γενική ιδέα, αναφέρεται σε ένα κατανεμημένο δίκτυο το οποίο συνδέει όλα τα «πράγματα» – αντικείμενα ή συσκευές – μέσω ασύρματων ετικετών και αισθητήρων με πρωτόκολλα δικτύων παρόμοια με εκείνα που χρησιμοποιούνται στο Διαδίκτυο. Τα «πράγματα» μπορούν να αναγνωριστούν αυτόματα, να επικοινωνήσουν μεταξύ τους, ακόμα και να λάβουν αποφάσεις από μόνα τους. Ειδικοί εκτιμούν ότι το ΔτΠ θα αποτελείται από 50-100 δισεκατομμύρια συσκευές έως το 2020. Λόγω του αυξανόμενου αριθμού διαθέσιμων αισθητήρων και της μεγάλης ποσότητας δεδομένων που παράγεται με πολύ μεγάλη ταχύτητα από το ΔτΠ, η ανάγκη για αποτελεσματικές και αποδοτικές μεθόδους για την διαχείριση αυτών των δεδομένων είναι αδιαμφισβήτητη. Συνεπώς, η εξόρυξη δεδομένων παίζει αποφασιστικής σημασίας ρόλο στο να γίνει το σύστημα αρκετά «έξυπνο» ώστε να μπορεί να παρέχει καταλληλότερες υπηρεσίες. Η ομαδοποίηση είναι μία από τις βασικές εργασίες της εξόρυξης δεδομένων και ένα ισχυρό εργαλείο ανάλυσης δεδομένων. Πρόκειται για μια διαδικασία η οποία στοχεύει στην οργάνωση ενός συνόλου δεδομένων εισόδου σε ένα σύνολο σημασιολογικά σύμφωνων ομάδων (ή συστάδων) με βάση ορισμένα μέτρα ομοιότητας, χωρίς καμία προηγούμενη γνώση, ώστε να εξάγει πολύτιμες πληροφορίες ανάλογα με τον στόχο της εκάστοτε εφαρμογής. Η παρούσα διπλωματική εργασία παρουσιάζει μια σύντομη επισκόπηση μερικών state-of-the-art αλγορίθμων ομαδοποίησης και κάνει ανασκόπηση της έρευνας που διεξήχθη σε αυτό το πεδίο από το 2010 έως και το 2015. Κατά τη διάρκεια αυτού του διαστήματος, το ΔτΠ προσέλκυσε το ενδιαφέρον των ερευνητών και δημοσιεύτηκαν πολυάριθμα επιστημονικά άρθρα σχετικά με την βελτίωση της απόδοσης και της ποιότητας, την μείωση της υπολογιστικής πολυπλοκότητας, καθώς και την ελαχιστοποίηση της κατανάλωσης ενέργειας στα Ασύρματα Δίκτυα Αισθητήρων (ΑΔΑ). Ωστόσο, δεδομένου ότι η τεχνολογία εξελίσσεται συνεχώς, είναι ξεκάθαρο ότι υπάρχουν περιθώρια βελτίωσης προς την βελτιστοποίηση των υφιστάμενων μεθόδων ή για ανάπτυξη νέων τεχνικών που μπορούν να εφαρμοστούν στο ΔτΠ.

Λέξεις κλειδιά: Διαδίκτυο των Πραγμάτων (ΔτΠ), ομαδοποίηση, αλγόριθμοι, εξόρυξη δεδομένων, μεγάλα δεδομένα

Abstract

The Internet of Things (IoT), as an emerging concept, refers to a distributed network connecting all "things" – objects or devices – through wireless tags and sensors over network protocols similar to those used in the Internet. Things can be identified automatically, communicate with each other and even make decisions by themselves. Experts estimate that the IoT will consist of 50-100 billion devices by 2020. Due to the increasing number of sensors available and huge amount of data generated at very high velocity by the IoT, there is an undeniable need for effective and efficient methods that are capable of handling these data. Therefore, data mining plays a critical role in making this kind of system smart enough to provide more convenient services. Clustering is one of the main tasks of data mining and a powerful data analysis tool. It is a process which aims to organize an input dataset into a set of semantically consistent groups, called clusters, with respect to some similarity measures, without any prior knowledge, in order to extract valuable information depending on each application's objective. The present diploma thesis displays a quick overview of some state-of-the-art clustering algorithms and reviews research conducted in this field between 2010 and 2015. The IoT has attracted much research attention during this period and numerous scientific papers have been published regarding performance and quality improvement, reduction of computational complexity, as well as minimization of energy consumption in Wireless Sensor Networks (WSNs). However, since technology is constantly evolving, it is clear that there is room for improvement towards optimization of the existing methods or for deployment of new techniques that can be applied to the IoT.

Keywords: Internet of Things (IoT), clustering, algorithms, data mining, big data

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στον Τομέα Συστημάτων Μετάδοσης Πληροφορίας και Τεχνολογίας Υλικών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.

Αρχικά, θα ήθελα να ευχαριστήσω τον Καθηγητή Διονύσιο – Δημήτριο Κουτσούρη για την ανάθεση του θέματος της παρούσας διπλωματικής εργασίας, καθώς στάθηκε αφορμή για να γνωρίσω μια πτυχή ενός πολύ σύγχρονου θέματος με εξαιρετικά ευρεία εφαρμογή.

Ευχαριστώ θερμά και τον Διδάκτορα Ιωάννη Κουρή, επιβλέποντα της εργασίας, για τον πολύτιμο χρόνο που αφιέρωσε στην καθοδήγησή μου.

Τέλος, ευχαριστώ ειλικρινά τον πατέρα μου και τον αδερφό μου για τη συνεχή υποστήριξη και την υπομονή που επέδειξαν κατά τη διάρκεια των σπουδών μου στο ΕΜΠ.

Αλόη – Αφροδίτη Αντωνίου

Αθήνα, Μάρτιος 2016

Περιεχόμενα

Κεφάλαιο 1 – Εισαγωγή

1.1 Στόχος της εργασίας

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανασκόπηση της έρευνας που πραγματοποιήθηκε την χρονική περίοδο 2010 έως και 2015 όσον αφορά τις τεχνικές ομαδοποίησης (clustering techniques) οι οποίες μπορούν να χρησιμοποιηθούν για την εξόρυξη δεδομένων από το Διαδίκτυο των Πραγμάτων. Οι τεχνικές ομαδοποίησης ποικίλλουν ως προς τον τρόπο προσέγγισης και τον τελικό στόχο του προβλήματος προς επίλυση. Γι' αυτό το λόγο, μια συνοπτική παρουσίαση της πιο πρόσφατης έρευνας σε αυτό το πεδίο είναι χρήσιμη για την εκτίμηση της προόδου και της κατεύθυνσης στην οποία πρέπει να επικεντρωθεί η μελλοντική έρευνα. Αξίζει να σημειωθεί πως, πέρα από τους state of the art αλγορίθμους, πολύ μεγάλο κομμάτι της έρευνας έχει πραγματοποιηθεί στο προαναφερθέν χρονικό διάστημα, και μάλιστα, χρόνο με το χρόνο οι δημοσιευμένες εργασίες πληθαίνουν.

1.2 Δομή της εργασίας

Στο δεύτερο κεφάλαιο, αρχικά, αναφέρεται η διαδικασία η οποία ακολουθήθηκε για την εύρεση των σχετικών με το θέμα δημοσιεύσεων. Στο τρίτο κεφάλαιο γίνεται μια γρήγορη επισκόπηση των κατηγοριών στις οποίες χωρίζονται οι τεχνικές ομαδοποίησης και στη συνέχεια παρουσιάζονται κάποιοι state of the art αλγόριθμοι για κάθε μία από αυτές τις κατηγορίες. Το τέταρτο κεφάλαιο αποτελεί το κυρίως μέρος. Σε αυτό αναλύονται οι νέες τεχνικές που προτάθηκαν στα επιστημονικά άρθρα αναφέροντας τις τεχνικές στις οποίες ενδεχομένως βασίστηκαν και εστιάζοντας στον τρόπο με τον οποίο βελτιώθηκαν, καθώς και στα αποτελέσματα της αξιολόγησής τους. Επίσης, παρατίθενται κάποιοι πίνακες που συνοψίζουν τα βασικά σημεία των τεχνικών. Στη συνέχεια, στο πέμπτο κεφάλαιο γίνεται μια ανασκόπηση των τεχνικών που αναλύθηκαν προηγουμένως, δίνοντας έμφαση στα αποτελέσματα των αξιολογήσεων. Τέλος, στον επίλογο γίνονται κάποιες γενικές παρατηρήσεις με βάση την προκειμένη αναζήτηση.

Κεφάλαιο 2

2.1 Μεθοδολογία

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανασκόπηση της έρευνας που πραγματοποιήθηκε την χρονική περίοδο 2010 έως και 2015 όσον αφορά τις τεχνικές ομαδοποίησης (clustering techniques) οι οποίες μπορούν να χρησιμοποιηθούν για την εξόρυξη δεδομένων από το Διαδίκτυο των Πραγμάτων. Η αναζήτηση των δημοσιεύσεων πραγματοποιήθηκε στις μηχανές αναζήτησης επιστημονικών άρθρων ScienceDirect (http://www.sciencedirect.com/),IEEE Xplore (http://ieeexplore.ieee.org/Xplore/home.jsp) και Pubmed (http://ncbi.nlm.nih.gov/pubmed). Οι λέξεις κλειδιά που χρησιμοποιήθηκαν σε κάθε περίπτωση ήταν οι εξής: “clustering”, “internet of things”, “algorithm”, με αναζήτησή τους κυρίως στον τίτλο, στο abstract και στις λέξεις κλειδιά των δημοσιεύσεων. Ύστερα από την εφαρμογή κάποιων φίλτρων για τον περιορισμό των αποτελεσμάτων, ελέγχθηκαν ένα προς ένα ως προς την σχετικότητά τους με το αντικείμενο της μελέτης.

Κεφάλαιο 3 – Τεχνικές ομαδοποίησης (Clustering Techniques)

Το 1854 στο Λονδίνο, κατά της διάρκεια ξεσπάσματος επιδημίας χολέρας, ο John Snow (Άγγλος γιατρός) χαρτογράφησε τα κρούσματα της ασθένειας που είχαν αναφερθεί [1]. Μετά τη δημιουργία του χάρτη, παρατήρησε ότι υπήρχε στενή σχέση ανάμεσα στην πυκνότητα των κρουσμάτων και ενός πηγαδιού που βρισκόταν σε κεντρικό δρόμο. Ο Snow έτσι σκέφτηκε πως η παροχή νερού μολυνόταν από λύματα και αυτό προκαλούσε την ταχεία εξάπλωση της ασθένειας. Ύστερα από αυτό, η συγκεκριμένη αντλία νερού αφαιρέθηκε και η επιδημία σταμάτησε. Συνήθως οι συσχετίσεις μεταξύ φαινομένων δεν είναι τόσο εύκολο να ανιχνευθούν, όμως το παραπάνω απλό παράδειγμα αποτελεί για πολλούς ερευνητές την πρώτη εφαρμογή της Ανάλυσης κατά Συστάδες (Cluster Analysis) ή Συσταδοποίησης/Ομαδοποίησης (Clustering). Έκτοτε, η Ανάλυση κατά Συστάδες έχει χρησιμοποιηθεί σε πολλούς τομείς και εφαρμογές, όπως η βιοπληροφορική, η ιατρική, το business και το μάρκετινγκ, το διαδίκτυο, η επιστήμη των υπολογιστών, αλλά και σε κοινωνικές επιστήμες, κ.ά. ώστε να αναγνωρίσουν «φυσικά» γκρουπ με κοινά χαρακτηριστικά ανάμεσα σε μεγάλες ποσότητες δεδομένων.

Στα πλαίσια του Διαδικτύου των Πραγμάτων (*Internet of Things*), όπου επιχειρείται η σύνδεση δισεκατομμυρίων αντικειμένων και συσκευών, ως προέκταση του ήδη υπάρχοντος διαδικτύου, η ποσότητα δεδομένων και πληροφοριών που θα παράγεται και θα μεταφέρεται είναι πρωτόγνωρα μεγάλη. Αυτό το νέο είδος δεδομένων ορίζεται ως *Big Data* («Μεγάλα Δεδομένα») και χαρακτηρίζεται κυρίως από τα λεγόμενα “3 V”: volume, variety, velocity – δηλαδή από τον μεγάλο όγκο των δεδομένων, την ποικιλία των ειδών των δεδομένων και την ταχύτητα με την οποία παράγονται. Τα εργαλεία ανάλυσης δεδομένων που είναι διαθέσιμα σήμερα δεν είναι αρκετά ισχυρά για την διαχείριση και ανάλυση των big data του Internet of Things. Έτσι, με την διάδοση αυτού του νέου concept, γίνεται όλο και πιο αναγκαία η εύρεση νέων αποτελεσματικών και αποδοτικών τεχνικών που θα μπορούν να εφαρμοστούν σε αυτό. Η Ομαδοποίηση (*Clustering*) είναι μία από τις βασικές εργασίες της Εξόρυξης Δεδομένων (*Data Mining*) και ένα ισχυρό εργαλείο ανάλυσης δεδομένων. Ο γενικός στόχος της διαδικασίας εξόρυξης δεδομένων είναι η εξαγωγή πληροφοριών από μεγάλες βάσεις δεδομένων και η μεταμόρφωσή τους σε μια κατανοητή δομή που μπορεί να φανεί χρήσιμη στους ανθρώπους.

3.1 Ορισμός Clustering

“Cluster analysis organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups. The representation can then be investigated to see if the data can be assigned to groups according to preconceived ideas or to suggest new experiments” [2].

Πιο απλά, *ομαδοποίηση* είναι η διαδικασία κατά την οποία δεδομένα κατατάσσονται σε σημασιολογικά σύμφωνες ομάδες (ή συστάδες) με βάση κάποιο μέτρο ομοιότητας, δηλαδή δεδομένα που ανήκουν στην ίδια ομάδα να είναι όμοια μεταξύ τους, ενώ δεδομένα από διαφορετικές ομάδες να είναι ανόμοια. Είναι ένα πρόβλημα μη επιβλεπόμενης μηχανικής μάθησης (unsupervised machine learning) που σημαίνει πως η δομή των δεδομένων πρέπει να ανιχνευτεί χωρίς να είναι διαθέσιμη κάποια άλλη προηγούμενη γνώση, ώστε να εξάγει πολύτιμες πληροφορίες ανάλογα με τον στόχο της εκάστοτε εφαρμογής. Πρόκειται για μια υποκειμενική διαδικασία, καθώς το ίδιο σύνολο δεδομένων πρέπει να διαχωριστεί διαφορετικά, ανάλογα την εφαρμογή. Σε αυτή την υποκειμενικότητα έγκειται και η δυσκολία της ομαδοποίησης, καθώς ένας αλγόριθμος ή μία συγκεκριμένη προσέγγιση δεν επαρκούν για να λύσουν κάθε πρόβλημα ομαδοποίησης. [3]

3.2 Κατηγοριοποίηση τεχνικών ομαδοποίησης

Πολλές διαφορετικές τεχνικές έχουν αναπτυχθεί, λοιπόν, με σκοπό να ανακαλύπτουν συνεκτικές ομάδες ανάμεσα σε μεγάλα σύνολα δεδομένων. Στη συνέχεια παρουσιάζονται δύο κλασικές κατηγορίες τεχνικών ομαδοποίησης, καθώς και κάποιες ειδικότερες.

3.2.1 Βασικές τεχνικές ομαδοποίησης

Διακρίνονται δύο βασικές κατηγορίες τεχνικών ομαδοποίησης: οι *Διαμεριστικοί* (*Partitional)* και οι *Ιεραρχικοί* (*Hierarchical) αλγόριθμοι ομαδοποίησης*. Οι ορισμοί τους είναι οι εξής [4]:

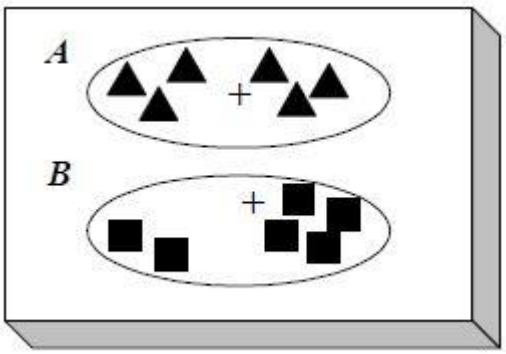
3.2.1.1 Διαμεριστικοί Αλγόριθμοι Ομαδοποίησης (Partitional Clustering Algorithms)

Ένας διαμεριστικός αλγόριθμος ομαδοποίησης κατασκευάζει χωρίσματα (partitions) της βάσης δεδομένων που του δίνεται, έτσι ώστε κάθε ομάδα (cluster) να ικανοποιεί ένα κριτήριο ομαδοποίησης, όπως πχ. η ελαχιστοποίηση του αθροίσματος των τετραγώνων της απόστασης από το μέσο (sum of squared distance from the mean) εντός κάθε cluster. Ένα πρόβλημα αυτών των αλγορίθμων είναι η

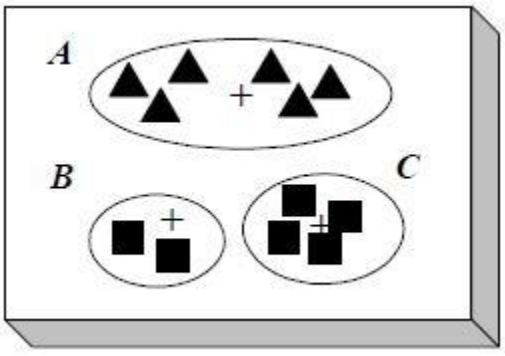
υψηλή πολυπλοκότητά τους, καθώς κάποιοι από αυτούς απαριθμούν όλους τους πιθανούς συνδυασμούς ομαδοποιήσεων και προσπαθούν να βρουν τον απόλυτα βέλτιστο. Ακόμα και για μικρό αριθμό δεδομένων, ο αριθμός των πιθανών χωρισμάτων είναι τεράστιος. Γι' αυτό συχνά τέτοια προβλήματα ξεκινούν με ένα αρχικό – συνήθως τυχαίο – partition και συνεχίζουν με την διόρθωσή του. Μια καλύτερη τακτική θα ήταν να τρέξει ο διαμεριστικός αλγόριθμος για διαφορετικά σετ k αρχικών σημείων (που θεωρούμε αντιπροσωπευτικά) και να διερευνηθεί κατά πόσον όλες οι λύσεις οδηγούν στο ίδιο τελικό partition.

Οι διαμεριστικοί αλγόριθμοι ομαδοποίησης προσπαθούν να βελτιώσουν ένα κριτήριο τοπικά. Πρώτα υπολογίζουν τις τιμές ομοιότητας ή απόστασης, ταξινομούν τα αποτελέσματα, και διαλέγουν εκείνο που βελτιστοποιεί το κριτήριο. Έτσι, οι περισσότεροι από αυτούς τους αλγορίθμους μπορούν να θεωρηθούν άπληστοι αλγόριθμοι (greedy-like algorithms).

Στα σχήματα 1 και 2 διακρίνονται δύο διαμεριστικοί αλγόριθμοι με αριθμό ομάδων (clusters) k=2 και k=3, αντίστοιχα. Το «+» υποδεικνύει το κέντρο των clusters, το οποίο στην προκειμένη περίπτωση ορίζεται ως ο μέσος όρος των τιμών ενός συγκεκριμένου cluster.



1. Διαμεριστικός αλγόριθμος με k=2



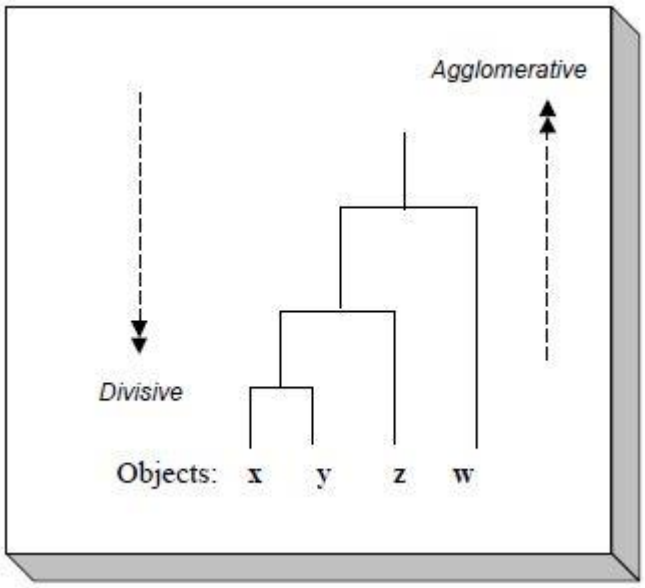
2. Διαμεριστικός αλγόριθμος με k=3

3.2.1.2 Ιεραρχικοί Αλγόριθμοι Ομαδοποίησης (Hierarchical Clustering Algorithms)

Οι ιεραρχικοί αλγόριθμοι ομαδοποίησης δημιουργούν μια ιεραρχική αποσύνθεση των δεδομένων. Είναι είτε συσσωρευτικοί (agglomerative/ bottom-up), είτε διαιρετικοί (divisive/ top-down):

- Οι ***συσσωρευτικοί*** αλγόριθμοι θεωρούν αρχικά κάθε στοιχείο ως έναν cluster και διαδοχικά συγχωνεύουν τους clusters σύμφωνα με κάποιο μέτρο εγγύτητας. Η ομαδοποίηση σταματά όταν όλα τα στοιχεία ανήκουν σε ένα γκρουπ ή όποτε το επιθυμεί ο χρήστης. Σε γενικές γραμμές, αυτές οι μέθοδοι ακολουθούν άπληστη συγχώνευση (greedy-like bottom-up merging).
- Οι ***διαιρετικοί*** αλγόριθμοι ακολουθούν την αντίθετη στρατηγική. Ξεκινούν με ένα γκρουπ που περιέχει όλα τα αντικείμενα (objects) και διαδοχικά «σπάει» σε μικρότερα γκρουπ έως ότου κάθε αντικείμενο αποτελεί τον δικό του cluster, ή όπως επιθυμεί ο χρήστης. Οι διαιρετικές προσεγγίσεις, σε κάθε βήμα, χωρίζουν τα αντικείμενα δεδομένων (data objects) σε γκρουπ χωρίς κοινά στοιχεία και ακολουθούν το ίδιο μοτίβο μέχρι όλα τα στοιχεία να βρίσκονται σε χωριστό cluster. Είναι παρόμοια λογική με αυτή που ακολουθούν οι «διαίρει και βασίλευε» αλγόριθμοι (divide-and-conquer algorithms).

Στο σχήμα 3 απεικονίζεται ένα δενδρόγραμμα που παράγεται είτε από έναν συσσωρευτικό είτε από έναν διαιρετικό ιεραρχικό αλγόριθμο:



3. Ιεραρχικός αλγόριθμος

Μια αδυναμία και των δύο προσεγγίσεων, είναι ότι άπαξ και εκτελεστεί μια συγχώνευση ή μια διαίρεση, δεν μπορεί να αναιρεθεί ή να βελτιωθεί. Οι διαμεριστικοί και οι ιεραρχικοί αλγόριθμοι μπορούν επίσης

να συνδυαστούν. Παραδείγματος χάριν, ένα αποτέλεσμα ιεραρχικής μεθόδου μπορεί να βελτιωθεί μέσω ενός διαμεριστικού βήματος το οποίο κάνει το αποτέλεσμα πιο «κομψό». Στη συνέχεια δίνονται και άλλες κατηγορίες αλγορίθμων ομαδοποίησης.

3.2.2 Τεχνικές Ομαδοποίησης για Εξόρυξη Δεδομένων (Data Mining Clustering Techniques)

Εκτός από τις δύο βασικές κατηγορίες των διαμεριστικών και ιεραρχικών αλγορίθμων ομαδοποίησης, έχουν αναπτυχθεί πολλές ακόμα μέθοδοι, οι οποίες εστιάζουν σε ειδικά προβλήματα ή ειδικά σύνολα δεδομένων. Αυτές οι μέθοδοι συμπεριλαμβάνουν τις εξής κατηγορίες [4]:

3.2.2.1 Ομαδοποίηση βάσει πυκνότητας (Density-Based Clustering)

Αυτοί οι αλγόριθμοι ομαδοποιούν στοιχεία σύμφωνα με ειδικές αντικειμενικές συναρτήσεις πυκνότητας (density objective functions). Η *πυκνότητα* ορίζεται ως ο αριθμός των στοιχείων που βρίσκονται σε μια συγκεκριμένη «γειτονιά» από στοιχεία. Σε αυτές τις τεχνικές το μέγεθος κάθε cluster αυξάνεται έως ότου η πυκνότητα, δηλαδή ο αριθμός των δεδομένων στη «γειτονιά», ξεπεράσει κάποιο κατώφλι. Ένας cluster που δημιουργείται με αυτόν τον τρόπο μπορεί να επεκτείνεται προς οπουδήποτε τον κατευθύνει η «πυκνότητα», γι' αυτό και με αυτό τον τύπο ομαδοποίησης μπορούν να δημιουργούνται clusters αυθαίρετης μορφής.

3.2.2.2 Ομαδοποίηση βάσει πλέγματος (Grid-Based Clustering)

Αυτοί οι αλγόριθμοι επικεντρώνονται σε χωρικά δεδομένα (spatial data), δηλαδή δεδομένα που μοντελοποιούν τη γεωμετρική δομή αντικειμένων στο χώρο, τις σχέσεις τους, τις ιδιότητες και τη λειτουργία τους. Πρώτα κβαντοποιούν το χώρο, δηλαδή χωρίζουν το σύνολο των δεδομένων σε κελιά, δημιουργώντας έτσι ένα «πλέγμα» και έπειτα δουλεύουν με βάση αυτό. Δεν μετακινούν σημεία, αλλά χτίζουν ιεραρχικά επίπεδα από σύνολα δεδομένων. Με αυτή την έννοια, είναι πιο κοντά στους ιεραρχικούς αλγορίθμους, όμως η συγχώνευση των κελιών και συνεπώς και των clusters, δεν εξαρτάται από κάποιο μέτρο απόστασης, αλλά από κάποια προκαθορισμένη παράμετρο. Πρόκειται για μια γρήγορη μέθοδο, αφού η ομαδοποίηση γίνεται μεταξύ των κελιών του πλέγματος και έτσι, τυπικά, η διαδικασία είναι ανεξάρτητη του αριθμού των δεδομένων.

3.2.2.3 Ομαδοποίηση βάσει μοντέλων (Model-Based Clustering)

Σε αυτή την κατηγορία, οι αλγόριθμοι βρίσκουν μια καλή προσέγγιση ενός μοντέλου παραμέτρων που ταιριάζει στα δεδομένα. Μπορούν να είναι είτε διαμεριστικοί είτε ιεραρχικοί, ανάλογα τη δομή ή το μοντέλο που θεωρούν για το σύνολο δεδομένων και τον τρόπο με τον οποίο βελτιστοποιούν τα μοντέλο ώστε να βρουν διχοτομήσεις (partitionings). Είναι πιο κοντά στους βασισμένους στην πυκνότητα (density-based) αλγορίθμους, καθώς αυξάνουν το μέγεθος συγκεκριμένων clusters έτσι ώστε να βελτιώνεται το υπάρχον μοντέλο. Παρόλα αυτά, κάποιες φορές ξεκινούν με έναν καθορισμένο αριθμό clusters και δεν χρησιμοποιούν την ίδια γενική ιδέα με την πυκνότητα.

Όπως έχει γίνει αντιληπτό ως τώρα, δεν υπάρχει μια τεχνική ομαδοποίησης που να ταιριάζει σε όλα τα προβλήματα. Κάποια μέθοδος μπορεί να λειτουργεί καλά για ένα σύνολο δεδομένων και ανεπαρκώς για κάποιο άλλο, ανάλογα το μέγεθος και τη διάσταση των δεδομένων, καθώς και την αντικειμενική συνάρτηση και τις δομές που χρησιμοποιήθηκαν. Παρόλα αυτά, ασχέτως απ' τη μέθοδο, υπάρχουν κάποια χαρακτηριστικά που οι ερευνητές θεωρούν πως περιγράφουν μια καλή τεχνική ομαδοποίησης. Πρόκεται σύμφωνα με το [4] για τα παρακάτω:

- ***Επεκτασιμότητα (Scalability)*:** Η ικανότητα του αλγορίθμου να αποδίδει καλά με μεγάλο αριθμό αντικειμένων δεδομένων.
- ***Ανάλυση ποικιλίας ειδών χαρακτηριστικών (Analyze mixture of attribute types):*** Η ικανότητα να αναλύει μοναδικά χαρακτηριστικά, αλλά και ποικιλία ειδών χαρακτηριστικών.
- ***Εύρεση ομάδων αυθαιρέτου σχήματος (Find arbitrary-shaped clusters):*** Το σχήμα συνήθως αντιστοιχεί στα είδη των clusters που ένας αλγόριθμος μπορεί να εντοπίσει και θα έπρεπε να θεωρείται πολύ σημαντική παράμετρος όταν επιλέγουμε μέθοδο, δεδομένου ότι θέλουμε να είμαστε όσο πιο γενικοί γίνεται.
- ***Ελάχιστες απαιτήσεις για τις παραμέτρους εισόδου (Minimum requirements for input parameters)*:** Πολλοί αλγόριθμοι ομαδοποίησης απαιτούν απ' τον χρήστη να ορίσει κάποιες παραμέτρους, όπως πχ. τον αριθμό των clusters που θα προκύψουν, ώστε να αναλυθούν τα δεδομένα. Παρόλα αυτά, με μεγάλα σύνολα δεδομένων και υψηλότερων διαστάσεων είναι επιθυμητό η μέθοδος να απαιτεί περιορισμένη καθοδήγηση απ΄ το χρήστη, προκειμένου να αποφευχθεί η ύπαρξη συστηματικού σφάλματος στο αποτέλεσμα.
- ***Αντιμετώπιση θορύβου (Handling of noise):*** Οι αλγόριθμοι ομαδοποίησης θα έπρεπε να μπορούν να χειριστούν αποκλίσεις, ώστε να βελτιώνουν την ποιότητα ομαδοποίησης. Ως

«αποκλίσεις» ορίζονται αντικείμενα δεδομένων που παρεκκλίνουν από τις κοινώς αποδεκτές νόρμες συμπεριφοράς και αναφέρονται και ως "*outliers*". Ο εντοπισμός των παρεκκλίσεων θεωρείται ξεχωριστό πρόβλημα.

- ***Ευαισθησία στην σειρά εισαγωγής καταγραφών (Sensitivity to the order of input records):*** Είναι δυνατό να παρουσιάσουμε το ίδιο σύνολο δεδομένων σε ορισμένους αλγορίθμους, αλλά με διαφορετική σειρά, και εκείνοι να παράγουν τελείως διαφορετικά αποτελέσματα. Η σειρά εισαγωγής επηρεάζει κυρίως αλγορίθμους που χρειάζονται μόνο ένα «πέρασμα» του συνόλου δεδομένων και δημιουργούν τοπικά βέλτιστες λύσεις σε κάθε βήμα. Γι' αυτό είναι πολύ σημαντικό οι αλγόριθμοι να μην είναι ευαίσθητοι στην σειρά εισαγωγής των στοιχείων του input.
- ***Δεδομένα υψηλών διαστάσεων (High dimensionality of data):*** Ο αριθμός των παραμέτρων (διαστάσεων) σε πολλά σύνολα δεδομένων είναι μεγάλος και πολλοί αλγόριθμοι ομαδοποίησης δεν μπορούν να χειριστούν περισσότερες από 8-10 διαστάσεις. Αποτελεί πρόκληση το να ομαδοποιηθούν σύνολα δεδομένων υψηλών διαστάσεων, όπως πχ. τα δεδομένα απογραφής των ΗΠΑ, που περιέχουν 138 παραμέτρους.
- ***Ερμηνεία και χρηστικότητα (Interpretability and usability):*** Είναι επιθυμητό τα αποτελέσματα που παράγουν οι αλγόριθμοι να ερμηνεύονται εύκολα ώστε να είναι και χρησιμοποιήσιμα.

Έχοντας αυτά τα χαρακτηριστικά κατά νου, παρακάτω παρουσιάζονται μερικοί από τους σημαντικότερους αλγορίθμους που έχουν επηρεάσει όσους ασχολούνται με την ομαδοποίηση.

3.2.3 Διαμεριστικοί Αλγόριθμοι (Partitional Algorithms)

Αυτή η οικογένεια αλγορίθμων ομαδοποίησης περιλαμβάνει τους πρώτους που εμφανίστηκαν στην κοινότητα της εξόρυξης δεδομένων. Εκείνοι που χρησιμοποιούνται πιο συχνά είναι οι: k*-means* [2][5], *PAM (Partitioning Around Medoids)* [5], *CLARA (Clustering LARge Applications)* [5] και *CLARANS (Clustering LARge ApplicatioNS)* [6].

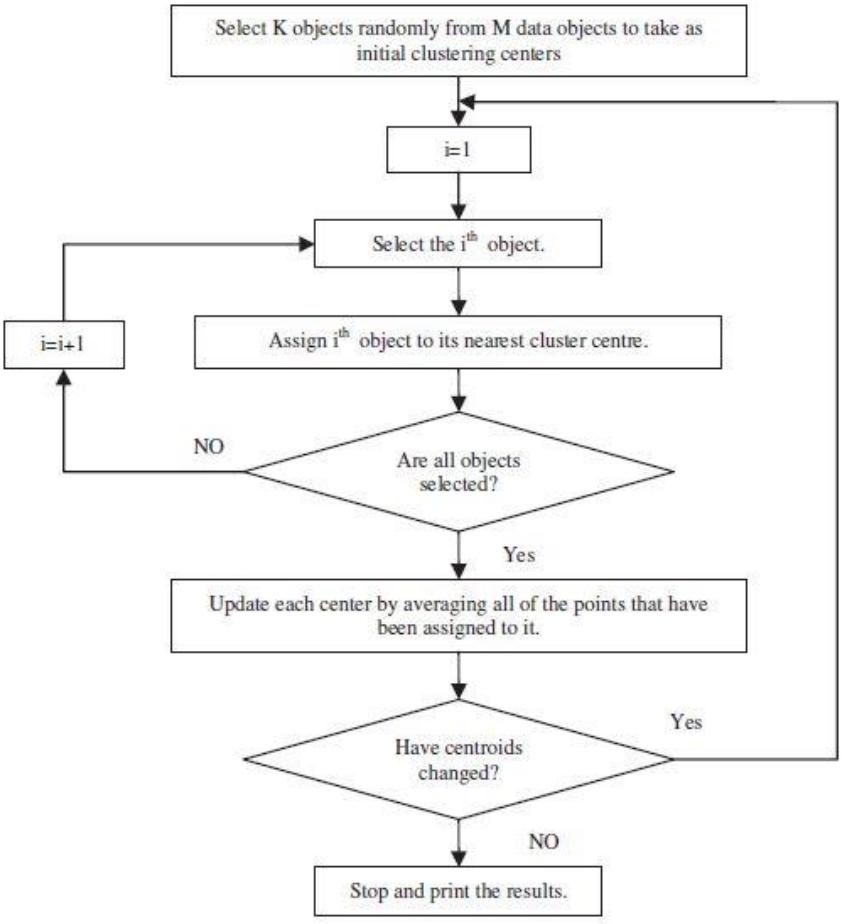
Ο σκοπός του ***k-means*** είναι να παράγει k clusters από ένα σύνολο n αντικειμένων, έτσι ώστε η αντικειμενική συνάρτηση τετραγωνικού σφάλματος (squared-error objective function):

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

να ελαχιστοποιηθεί. Στην παραπάνω έκφραση, C_i είναι οι clusters, p είναι ένα σημείο σε έναν cluster C_i και m_i είναι το κεντροειδές (centroid) του cluster C_i. Το κέντρο ενός cluster είναι το μέσο σημείο στον πολυδιάστατο χώρο που ορίζεται από τις διαστάσεις. Κατά μία έννοια είναι το κέντρο βαρύτητας του cluster και δεν αποτελεί απαραίτητα υπαρκτό σημείο – μπορεί δηλαδή να είναι νοητό. Ο χρήστης πρέπει να δώσει τον αριθμό των clusters, k, και ο αλγόριθμος επιστρέφει τα κέντρα (ή μέσα) του κάθε cluster C_i. Το μέτρο εγγύτητας που χρησιμοποιείται συνήθως είναι η Ευκλείδεια απόσταση. Πιο αναλυτικά, ο αλγόριθμος αποτελείται από τα παρακάτω βήματα:

1. Επίλεξε k αντικείμενα ως αρχικά κέντρα των clusters
2. Αντιστοίχισε κάθε αντικείμενο δεδομένων στο κοντινότερο κέντρο, σύμφωνα με το μέτρο εγγύτητας που χρησιμοποιείται
3. Υπολόγισε εκ νέου το κέντρο k για κάθε cluster
4. Επανάλαβε τα βήματα 2 και 3 μέχρι τα κέντρα να μην μετακινούνται άλλο.

Το διάγραμμα ροής του k-means φαίνεται στην εικόνα 4.



4. Διάγραμμα ροής του k-means αλγορίθμου

Τα μειονεκτήματα του k-means είναι τα εξής:

- Επεκτασιμότητα: Δεν αποδίδει καλά με μεγάλο αριθμό αντικειμένων δεδομένων.
- Αρχικά μέσα: Το αποτέλεσμα ομαδοποίησης είναι εξαιρετικά ευαίσθητο στα αρχικά μέσα.
- Θόρυβος: Ο θόρυβος ή outliers αλλοιώνει την ποιότητα του αποτελέσματο ομαδοποίησης.
- Αριθμός ομάδων: Ο αριθμός των clusters πρέπει να προσδιοριστεί προτού ξεκινήσει η ομαδοποίηση.
- Τοπικά ελάχιστα: Συγκλίνει συχνά σε τοπικά ελάχιστα καθώς η αντικειμενική του συνάρτηση είναι μη κυρτή.
- Αδυναμία ομαδοποίησης μη-γραμμικά διαχωρίσιμων συνόλων δεδομένων: Αποτυγχάνει να χωρίσει μη-γραμμικά διαχωρίσιμα σύνολα δεδομένων στον χώρο εισόδου.

Ο αλγόριθμος ***PAM (Partitioning Around Medoids)*** είναι μια προέκταση του k-*means* που προορίζεται για να χειρίζεται αποτελεσματικά τα outliers. Η διαφορά τους έγκειται στο ότι ο *PAM* αντί για κέντρα των clusters επιλέγει κάθε cluster να αντιπροσωπεύεται από το "medoid" του (ένα αντικείμενο με τη μικρότερη μέση ανομοιότητα προς όλα τα υπόλοιπα αντικείμενα του συνόλου). Το medoid είναι το object του cluster που βρίσκεται πιο κεντρικά – αυτή τη φορά δηλαδή δεν μπορεί να είναι κάποιο νοητό σημείο. Κατά συνέπεια, τα medoids δεν επηρεάζονται τόσο από ακραίες τιμές. Το μέσο ενός αριθμού αντικειμένων θα έπρεπε να «ακολουθήσει» αυτές τις ακραίες τιμές, ενώ το medoid όχι. Ο αλγόριθμος επιλέγει *k* αρχικά medoids και τοποθετεί άλλα αντικείμενα σε clusters των οποίων το medoid είναι πιο κοντά σε αυτά, ενώ εξακολουθεί να ανταλλάσσει medoids με μη-medoids όσο το αποτέλεσμα βελτιώνεται. Η ποιότητα μετριέται με το τετραγωνικό σφάλμα ανάμεσα στα αντικείμενα ενός cluster και του medoid του. Η υπολογιστική πολυπλοκότητα του *PAM* είναι $O(Ik(n-k)^2)$, όπου I είναι ο αριθμός των επαναλήψεων, οπότε πρόκειται για «δαπανηρό» αλγόριθμο για μεγάλες τιμές n και k.

Μια λύση σε αυτό είναι ο αλγόριθμος ***CLARA (Clustering LARge Applications),*** ο οποίος εφαρμόζει τον *PAM* σε κάθε δείγμα μεγέθους *s* των n πλειάδων (tuples) της βάσης δεδομένων. Το output εξαρτάται από τα *s* δείγματα και έχει αποδειχτεί ότι ο *CLARA* λειτουργεί καλά με 5 δείγματα μεγέθους $40 + k$ και η πολυπλοκότητά του γίνεται $O(k(40+k)^2 + k(n-k))$. Πάντως, υπάρχει ένα ζήτημα ποιότητας όταν χρησιμοποιούμε τεχνική δειγματοληψίας σε ομαδοποίηση: το αποτέλεσμα μπορεί να μην αντιπροσωπεύει το αρχικό σύνολο δεδομένων, αλλά μια τοπικά βέλτιστη λύση. Για παράδειγμα, αν τα αρχικά medoids δεν περιέχονται στο δείγμα, τότε το αποτέλεσμα δεν θα είναι το καλύτερο.

Ο ***CLARANS (Clustering LARge ApplicatioNS)*** συνδυάζει κι αυτός τον *PAM* με δειγματοληψία. Πιο συγκεκριμένα, η ομαδοποίηση γίνεται με αναζήτηση σε έναν γράφο: οι κόμβοι του γράφου είναι πιθανές λύσεις, δηλαδή ένα σύνολο k medoids. Η πολυπλοκότητα του *CLARANS* είναι $O(kn^2)$, γεγονός που δεν τον καθιστά κατάλληλο για μεγάλα σύνολα δεδομένων.

3.2.4 Ιεραρχικοί Αλγόριθμοι (Hierarchical Algorithms)

Οι καθιερωμένοι ιεραρχικοί αλγόριθμοι έχουν μεγάλη υπολογιστική πολυπλοκότητα, $O(n^2)$. Έχουν γίνει προσπάθειες βελτίωσης της απόδοσής τους και μία από τις πρώτες είναι ο αλγόριθμος ***BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)*** [7]. Για κάθε cluster, ο *BIRCH* αποθηκεύει στη μνήμη μόνο την τριπλέτα *(n, LS, SS)*, όπου *n* είναι ο αριθμός των αντικειμένων δεδομένων στον cluster, *LS* είναι το γραμμικό άθροισμα των τιμών των αντικειμένων του cluster και *SS* είναι το άθροισμα των τετραγώνων των τιμών των objects του cluster. Αυτές οι τριπλέτες ονομάζονται *Cluster Features (CF)* και τοποθετούνται σε ένα *CF-tree.* Το *CF-tree* είναι ένα ισοζυγισμένο δέντρο (height-balanced tree – AVL tree), το οποίο αποθηκεύει αυτά τα χαρακτηριστικά ομαδοποίησης και χαρακτηρίζεται από δύο παραμέτρους: τον *Παράγοντα Διακλάδωσης (Branching Factor)* B, και το *Κατώφλι (Threshold)* T. Το B είναι ο μέγιστος αριθμός παιδιών ενός εσωτερικού κόμβου (όχι φύλλο) και το T είναι η μέγιστη απόσταση μεταξύ οποιουδήποτε ζεύγους σημείων, δηλαδή η διάμετρος κάθε cluster στους κόμβους-φύλλα. Ο *BIRCH* μπορεί να βρει μια καλή ομαδοποίηση με ένα μόνο πέρασμα των δεδομένων (και να τα βελτιώσει με ένα προαιρετικό πέρασμα), οπότε η πολυπλοκότητά του είναι $O(n)$. Η αδυναμία του βρίσκεται στην ποιότητα των clusters που προκύπτουν. Πρώτον, εφόσον χρησιμοποιεί την Ευκλείδεια απόσταση, δουλεύει καλά μόνο σε καλά διανεμημένα αριθμητικά δεδομένα, και δεύτερον, η παράμετρος *T* επηρεάζει το μέγεθος των clusters και επομένως και την «φυσικότητά» τους, οπότε κάποιες φορές αναγκάζει αντικείμενα που θα έπρεπε να ανήκουν στο ίδιο cluster να ενταχθούν σε διαφορετικό, μόνο και μόνο επειδή «έφτασαν» με διαφορετική σειρά και ο κάθε κόμβος στο *CF-tree* μπορεί να κρατήσει έναν περιορισμένο αριθμό καταχωρήσεων λόγω του μεγέθους του. Κατά τα άλλα, ήταν ο πρώτος αλγόριθμος ομαδοποίησης που κατάφερε να χειριστεί τον θόρυβο αποτελεσματικά.

Ο αλγόριθμος ***CURE (Clustering Using Representatives*)** [8] χρησιμοποιεί περισσότερα από ένα αντιπροσωπευτικά σημεία σε κάθε cluster για να βελτιώσει την ποιότητα του αποτελέσματος και έτσι μπορεί να «συλλάβει» clusters διαφόρων σχημάτων και μεγεθών, σε μεγάλες βάσεις δεδομένων. Η πολυπλοκότητά του είναι $O(n^2 \log n)$. Είναι αξιόπιστη μέθοδος για clusters αυθαιρέτου σχήματος και

έχει καλή επίδοση σε δισδιάστατα σύνολα δεδομένων. Είναι, όμως, ευαίσθητος σε παραμέτρους όπως ο αριθμός των αντιπροσωπευτικών αντικειμένων, το συντελεστή συστολής (shrinkage factor) και τον αριθμό των partitions.

Γενικά, ο *BIRCH* αποδίδει καλύτερα από τον *CURE* σε χρονική πολυπλοκότητα, αλλά πάσχει στην ποιότητα ομαδοποίησης. Τα outliers τα χειρίζονται και οι δύο καλά.

3.2.5 Αλγόριθμοι βασισμένοι στην Πυκνότητα (Density-Based Algorithms)

Ο πιο δημοφιλής density-based αλγόριθμος είναι ο ***DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*** [9]. Ο DBSCAN ξεκινάει με ένα τυχαίο σημείο p και υπολογίζει την πυκνότητά του, δηλαδή τον αριθμό των σημείων στην ε-γειτονιά του p. Αν το p είναι σημείο πυρήνα (core point), ο DBSCAN σημειώνει αυτό το σημείο ως νέο cluster και ύστερα ανακτά όλα τα σημεία που είναι προσπελάσιμα βάσει πυκνότητας (density reachable) από το p και τους αναθέτει την ίδια cluster ετικέτα με του p. Αλλιώς, το σημείο p επισημαίνεται ως σημείο θορύβου (noisy point). Έπειτα, ο DBSCAN συλλέγει επαναληπτικά τα σημεία που είναι density reachable από τα core points. Η διαδικασία τερματίζει όταν δεν μπορούν να προστεθούν άλλα νέα αντικείμενα σε κανέναν cluster. Εντωμεταξύ, εάν κάποιο σημείο δεν βρίσκεται στην ε-γειτονιά κάποιου cluster, θεωρείται ως σημείο θορύβου, το οποίο είναι πιθανό outlier. Έτσι, ο DBSCAN εντοπίζει clusters αυθαιρέτου σχήματος και δεν επηρεάζεται από τη σειρά εισόδου των στοιχείων. Κάθε νέο στοιχείο που εισάγεται επηρεάζει μία μόνο γειτονιά.
Τα μειονεκτήματα του DBSCAN είναι τα εξής [10]:

- Η απόδοση της ομαδοποίησης εξαρτάται από δύο παραμέτρους, την ε ή Eps που συμβολίζει την μέγιστη ακτίνα της γειτονιάς από το σημείο παρατήρησης και από την $MinPts$ η οποία συμβολίζει τον ελάχιστο αριθμό αντικειμένων που πρέπει αυτή να περιέχει. Αυτές οι δύο παράμετροι καθορίζονται από το χρήστη, οπότε είναι δύσκολο να προσδιοριστούν οι κατάλληλες τιμές τους χωρίς καμία πρότερη γνώση.
- Η πολυπλοκότητά του είναι υψηλή όταν αντιμετωπίζει σύνολα δεδομένων υψηλών διαστάσεων. Η πολυπλοκότητα του DBSCAN είναι $O(n^2)$ όταν δεν εφαρμόζεται κάποια δομή δεικτοδότησης (indexing structure) λόγω του υπολογισμού του μέτρου ομοιότητας μεταξύ σημείων δεδομένων, παρόλο που σαρώνει ολόκληρο το σύνολο δεδομένων μόνο μία φορά. Αυτό σημαίνει ότι ο DBSCAN έχει προβλήματα επεκτασιμότητας όταν εφαρμοστεί σε μεγάλο σύνολο δεδομένων. Αν

χρησιμοποιηθούν αποτελεσματικές δομές δεικτοδότησης και οι διαστάσεις των σημείων είναι χαμηλές ($d \leq 5$) τότε η υπολογιστική πολυπλοκότητα του DBSCAN μειώνεται σε $O(n \log n)$.

- Είναι ευαίσθητος στην σειρά με την οποία εισάγονται τα δεδομένα. Οπότε διαφορετική σειρά των σημείων του ίδιου συνόλου δεδομένων καταλήγουν σε διαφορετικά αποτελέσματα.
- Παρακείμενοι clusters διαφορετικών πυκνοτήτων μπορεί να μην αναγνωριστούν σωστά.

Ο ***OPTICS (Ordering Points to Identify the Clustering Structure)*** [11] είναι προέκταση του *DBSCAN* και δεν έχει τόσο αυστηρές απαιτήσεις για τις παραμέτρους εισόδου. Διατάσσει γραμμικά τα στοιχεία της βάσης δεδομένων, έτσι ώστε σημεία που βρίσκονται χωρικά πιο κοντά, γίνονται γείτονες σε αυτή τη διάταξη, και ύστερα ομαδοποιεί αυτόματα τα δεδομένα. Η υπολογιστική πολυπλοκότητά του είναι ίδια με του *DBSCAN*, $O(n \log n)$.

Κεφάλαιο 4 – Πέρα από την αιχμή της επιστήμης (beyond the state of the art)

4.1 Σύνοψη

Οι δημοσιεύσεις που συγκεντρώθηκαν από τις μηχανές αναζήτησης επιστημονικών άρθρων χωρίστηκαν σε τέσσερις κατηγορίες σύμφωνα με τον σκοπό για τον οποίο αναπτύχθηκαν οι τεχνικές που προτείνουν. Οι κύριοι στόχοι είναι οι εξής:

- Βελτίωση της απόδοσης
- Μείωση της κατανάλωσης ενέργειας στα WSNs
- Μείωση της πολυπλοκότητας
- Βελτίωση της ποιότητας

Τα βασικά σημεία των δημοσιεύσεων συνοψίζονται ανά κατηγορία στους παρακάτω πίνακες:

Στόχος: Βελτίωση απόδοσης

Τίτλος δημοσίευσης	Στόχος	Νέα τεχνική/ Βελτίωση υπάρχουσας	Τεχνικές στις οποίες έχει βασιστεί	Αποτελέσματα αξιολόγησης
A Fast Density-Based Clustering Algorithm for Real-Time Internet of Things Stream *(Hybrid Density-based Clustering for data stream (HDC-Stream) algorithm)*	Βελτίωση απόδοσης (ποιότητα των αποτελεσμάτων, χρόνος υπολογισμού)	Βελτίωση υπάρχουσας/ Υβριδική μέθοδος	Density grid-based clustering Density microclustering Modified DBSCAN	HDC-Stream: υψηλή ποιότητα με χαμηλό χρόνο υπολογισμού for merging Καλύτερη απόδοση από τους DenStream και D-Steam Ακατάλληλος για διανεμημένα περιβάλλοντα
A Hybrid Approach to Clustering in Big Data *(νέος clusiVAT)*	Βελτίωση CPU time και partition accuracy (PA)	Βελτίωση υπάρχουσας	VAT αλγόριθμοι Modified Prim's algorithm	Νέος clusiVAT: μεγαλύτερη ταχύτητα και ακρίβεια από τους: k-means, single pass k-means(spkm), online k-means(okm), and clustering using representatives (CURE) Κατάταξη με βάση την ακρίβεια, από τον καλύτερο στον χειρότερο: νέος clusiVAT, CURE, spkm, k-means, and okm. Κατάταξη με βάση τον χρόνο εκτέλεσης, από τον γρηγορότερο στον πιο αργό: νέος clusiVAT, k-means, okm, spkm, and CURE
A new credibilistic clustering algorithm *(Credibilistic Clustering Model (CCM))*	Βελτίωση απόδοσης	Βελτίωση υπάρχουσας	Possibilistic C-Means(PCM)	CCM: πιο αποδοτικός από τους PCM (Possibilistic C-Means) και FCM (Fuzzy C-Means) - εξαλείφει τα προβλήματά τους
A new mechanism for RFID clustering and identification	Βελτίωση απόδοσης της διαδικασίας αναγνώρισης ετικετών (tags) και συλλογής δεδομένων σε RFID συστήματα (μείωση χρόνου αναγνώρισης, αποφυγή "σύγκρουσης" των ετικετών)	Βελτίωση υπάρχουσας	Progressive Scanning (PS) Algorithm	Καλύτερη απόδοση από τους EDFSA (Enhanced Dynamic Framed Slotted Aloha) και DFSA (Dynamic Framed Slotted Aloha) από άποψη απόδοσης και χρόνου για την αναγνώριση ετικετών

An efficient and scalable density-based clustering algorithm for datasets with complex structures *(Influence Space and Detection of borderpoints(ISB-DBSCAN algorithm))*	Διόρθωση των συνεπειών των (κακών) αρχικοποιήσεων	Βελτίωση υπάρχουσας	DBSCAN	ISB-DBSCAN: καλύτερη απόδοση από τον DBSCAN και τον IS-DBSCAN. Επιταχύνει τον DBSCAN και ενισχύει την απόδοσή του στην διάκριση παρακείμενων clusters με διαφορετικές πυκνότητες και στην ανίχνευση οριακών σημείων Μειώνει τις απαιτούμενες παραμέτρους και την ευαισθησία στα σημεία εκκίνησης Καταλληλότερος για σύνολα δεδομένων πολλών διαστάσεων
Bias-correction fuzzy clustering algorithms	Βελτίωση απόδοσης	Βελτίωση υπάρχουσας	Fuzzy C-Means (FCM) Generalized FCM (GFCM)	Σε γενικές γραμμές, η μέθοδος βελτίωσε τα αποτελέσματα
Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm *(GAKFCM - combination: improved genetic algorithm & the kernel technique)*	Βελτίωση απόδοσης του Fuzzy C-means	Βελτίωση υπάρχουσας	Fuzzy C-Means(FCM) Genetic Algorithm(GA) Kernel-based fuzzy c-means (KFCM)	GAKFCM: εξαλείφει τα ελαττώματα του FCM και βελτιώνει σε μεγάλο βαθμό την απόδοση
An Adaptive Meta-Heuristic Search for the Internet of Things *(AntClust)*	Βελτίωση της απόδοσης της context-aware αναζήτησης αισθητήρων στο IoT	Βελτίωση υπάρχουσας	Ant clustering	AntClust: σημαντικά ταχύτερος από τον CASSARAM, αλλά με ελαφρώς μικρότερη ακρίβεια Επεκτάσιμος
An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering *(Hybrid K-MICA)*	Αποφυγή σύγκλισης του k-means σε τοπικά βέλτιστα	Βελτίωση υπάρχουσας/ Hybrid method	k-means Modified Imperialist Competitive Algorithm(MICA)	Hybrid K-MICA: Βιώσιμη και αποδοτική μέθοδος Συγκρίσιμος με τους MICA, ICA, ACO, PSO, SA, GA, TS, HBMO και k-means Μπορεί να χρησιμοποιηθεί μόνο όταν ο αριθμός των clusters είναι γνωστός εκ των προτέρων
Clustering Massive Small Data for IOT	Βελτίωση απόδοσης του συστήματος Hadoop	Βελτίωση υπάρχουσας	K-means	Βελτίωση της απόδοσης επεξεργασίας δεδομένων και αύξηση του ποσοστού χρησιμοποίησης των πόρων του συστήματος
Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion *(Web Document Clustering based on the Cuckoo Search Algorithm (WDC-CSK))*	Βελτίωση απόδοσης στην ομαδοποίηση αποτελεσμάτων στο διαδίκτυο	Βελτίωση υπάρχουσας	Cuckoo search (CS) meta-heuristic algorithm k-means algorithm Balanced Bayesian Information Criterion	WDC-CSK: βελτίωση ακρίβειας αποτελεσμάτων σε σύγκριση με τους Suffix Tree Clustering (STC), Lingo και Bisecting k-means
GGSA: A Grouping Gravitational Search Algorithm for data clustering (Grouping Gravitational Search Algorithm - GGSA)	Βελτίωση απόδοσης	Βελτίωση υπάρχουσας	Gravitational Search Algorithm (GSA)	GGSA: πιο ακριβής από τους PSO και GSA Πολύ καλή απόδοση σε σύγκριση με τους Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Firefly Algorithm (FA) και εννέα ακόμα γνωστές τεχνικές ομαδοποίησης

Στόχος: Βελτίωση ποιότητας

Τίτλος δημοσίευσης	Στόχος	Νέα τεχνική/ Βελτίωση υπάρχουσας	Τεχνικές στις οποίες έχει βασιστεί	Αποτελέσματα αξιολόγησης
A Cloud-Friendly RFID Trajectory Clustering Algorithm in Uncertain Environments *(HRTC algorithm)*	Βελτίωση ποιότητας σε εφαρμογές ιχνηλασιμότητας με RFID	Νέα τεχνική	Χρήση του OPTICS	HRTC: είναι επεκτάσιμος και πιο αποδοτικός από τους Time-Focused Clustering (TFC) και Fuzzy C-Means (FCM) από άποψη ποιότητας, όταν ο αριθμός των clusters είναι μεγάλος (ποιότητα = Sum of Squared Error (SSE), μεγάλος = μεγαλύτερος από 70) Λιγότερο ευαίσθητος σε θόρυβο και outliers
A New Data Clustering Algorithm *(Induction and Deduction Clustering (IDC) algorithm)*	Βελτίωση ποιότητας αποτελεσμάτων	Νέα τεχνική	-	IDC: μεγαλύτερη ακρίβεια και ικανότητα ανίχνευσης outliers από τους BIRCH και k-means
Automatic kernel clustering with bee colony optimization algorithm *(AKC-BCO)*	Βελτίωση ποιότητας με αυτόματη ομαδοποίηση (Αποφυγή της απαίτησης ορισμού του αριθμού των clusters εκ των προτέρων)	Βελτίωση υπάρχουσας	Bee colony optimization (BCO) algorithm Kernel function	AKC-BCO: μεγαλύτερη ακρίβεια από τους Dynamic Clustering using the binary-Particle Swarm Optimization (DCPSO), Dynamic Clustering based on PSO and Genetic algorithm (DCPG) και Automatic Kernel Clustering with Multi-Elitist PSO (AKC-MEPSO), αλλά με περισσότερο υπολογιστικό χρόνο

Στόχος: Μείωση κατανάλωσης ενέργειας

Τίτλος δημοσίευσης	Στόχος	Νέα τεχνική/ Βελτίωση υπάρχουσας	Τεχνικές στις οποίες έχει βασιστεί	Αποτελέσματα αξιολόγησης
An Energy Balanced Cluster Algorithm for Wireless Sensor Networks	Βελτίωση ενεργειακής απόδοσης	Νέα τεχνική	-	Καλύτερη κατανομή του ενεργειακού φόρτου και παράταση της ζωή του WSN σε σχέση με τους "Opt" και Deterministic Cluster-Head Selection (DCHS) αλγορίθμους
An energy efficient hierarchical clustering index tree for facilitating time-correlated region queries in the Internet of Things *(ECH-tree)*	Βελτίωση ενεργειακής απόδοσης	Βελτίωση υπάρχουσας	Grid cell clustering	ECH-tree: καλύτερη απόδοση και παράταση της ζωής του δικτύου, ειδικά σε περιοχές με πυκνή τοποθέτηση των αισθητήρων
Design of an Improved Energy Efficient Clustering in M2M Communication *(Improved M2M clustering process (IMCP) method)*	Βελτίωση ενεργειακής απόδοσης	Βελτίωση υπάρχουσας	Χρησιμοποιεί τη δημοσίευση "Energy-Efficient Clustering Design for M2M Communications"	IMCP: πιο σύνθετος, αλλά παρατείνει τη διάρκεια ζωής του δικτύου σε σχέση με τον LEACH και τον Energy Aware Multi-Hop Multi-Path Hierarchy (EAMMH)
NDCMC: A Hybrid Data Collection Approach for Large-Scale WSNs Using Mobile Element and Hierarchical Clustering *(Node Density based Clustering and Mobile Collection)*	Βελτίωση ενεργειακής απόδοσης	Βελτίωση υπάρχουσας/ Hybrid approach (NDCMC) + Νέα τεχνική: Random Clustering and Mobile Collection (RCMC)	Συνδυασμός Hierarchical routing & συλλογής δεδομένων με κινητά στοιχεία (mobile elements - ME)	NDCMC: βελτιώνει την διάρκεια ζωής του δικτύου Καλύτερη ενεργειακή απόδοση από τους MILP, CSPLI και SST (μέθοδοι με MEs) και τον LEACH
Service-Aware Clustering An Energy-Efficient Model for the Internet-of-Things *(SAC protocol/algorithm)*	Βελτίωση ενεργειακής απόδοσης	Νέα τεχνική	-	SAC: μεγαλύτερη διάρκεια ζωής του δικτύου από έναν Breadth-First Search (BFS) και έναν Service-Blind Clustering (SBC) αλγόριθμο, τον LEACH, τον LEACH-C και τους πιο πρόσφατους DECSA και MOCRN
A Density-based Energy-efficient Clustering Algorithm for Wireless Sensor Networks *(DECA)*	Βελτίωση ενεργειακής απόδοσης	Νέα τεχνική	-	DECA: ομοιόμορφη κατανομή των cluster-heads και παράταση της ζωής του δικτύου σε σχέση με τους LEACH και Density-based Clustering Protocol (DBCP).

Στόχος: Μείωση πολυπλοκότητας

Τίτλος δημοσίευσης	Στόχος	Νέα τεχνική/ Βελτίωση υπάρχουσας	Τεχνικές στις οποίες έχει βασιστεί	Αποτελέσματα αξιολόγησης
Fast modified global k-means algorithm for incremental cluster construction (*Fast Modified Global k-means - FMGKM)*	Μείωση υπολογιστικής προσπάθειας και χρήσης της μνήμης	Βελτίωση υπάρχουσας	k-means Global k-means Modified global k-means	FMGKM: ταχύτερος και ακριβέστερος από τον GKM, στις περισσότερες περιπτώσεις παρόμοια αποτελέσματα με τον MGKM, αλλά σε λιγότερο χρόνο επεξεργασίας (CPU time)
An agglomerative clustering algorithm using a dynamic k-nearest-neighbor list *(DKNNA+FS)*	Μείωση υπολογιστικής πολυπλοκότητας	Νέα τεχνική	Γίνεται χρήση του Fast search algorithm	DKNNA: πετυχαίνει σχεδόν τα ίδια αποτελέσματα με του FPNN Σε σχέση με τον FPNN με FS μειώνει τον χρόνο υπολογισμού
An efficient hyperellipsoidal clustering algorithm for resource-constrained environments *(HyCARCE)*	Χαμηλή υπολογιστική πολυπλοκότητα	Νέα τεχνική	-	HyCARCE: έχει συγκρίσιμη υπολογιστική πολυπλοκότητα με τους DENCLUE, k-means και Gustafson–Kessel(GK), αλλά καλύτερη από του subtractive clustering(SC) Ταχύτερος και λιγότερο ευαίσθητος στις παραμέτρους εισόδου από τον DENCLUE Η ακρίβεια των αποτελεσμάτων επηρεάζεται σε υψηλότερες διαστάσεις, άρα καταλληλότερος για δεδομένα χαμηλών διαστάσεων
A new topological clustering algorithm for interval data	Ομαδοποίηση συνεχών δεδομένων ή δεδομένων διαστήματος (interval data) με χαμηλό υπολογιστικό κόστος	Βελτίωση υπάρχουσας	Self-Organizing Map (SOM) S2L-SOM (Simultaneous Two-Levels-SOM)	Παρόμοια ή καλύτερα αποτελέσματα από τους DIV, SCLUST, SHICLUST, SYKSOM, SYKCLUST σε μικρότερο χρόνο επεξεργασίας
A time-efficient pattern reduction algorithm for k-means clustering *(Pattern Reduction (PR) algorithm)*	Μείωση χρόνου υπολογισμού	Βελτίωση υπάρχουσας	k-means	PR: μειώνει σημαντικά τον χρόνο υπολογισμού του k-means και των βασισμένων στον k-means αλγορίθμων
Fuzzy joint points based clustering algorithms for large data sets *(OFJP και aScan)*	Μείωση χρόνου υπολογισμού	Βελτίωση υπάρχουσας	FJP	OFJP και aScan: δραματικά γρηγορότεροι από τον modified FJP (MFJP) Κατάλληλος για μεγάλα σύνολα δεδομένων

4.2 Στόχος: Βελτίωση απόδοσης

4.2.1 A Fast Density-Based Clustering Algorithm for Real-Time Internet of Things Stream

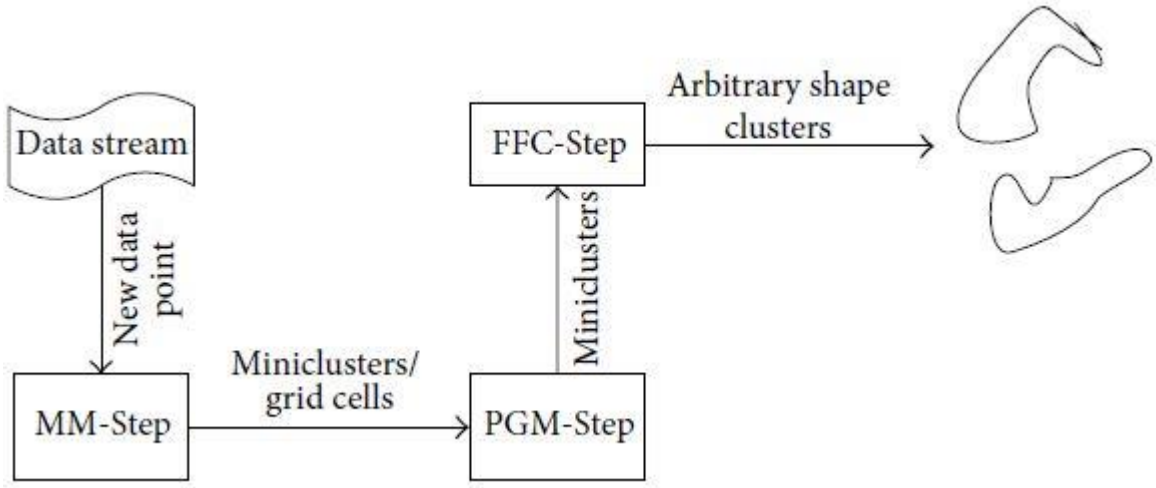
Οι συσκευές του Διαδικτύου των Πραγμάτων παράγουν αδιάλειπτα ροές δεδομένων (data streams) με την πάροδο του χρόνου. Έτσι, ο όγκος δεδομένων που συλλέγονται από RFID (Radio-frequency identification) και συμβατικούς αισθητήρες, είναι πραγματικά μεγάλος και απαιτούνται «έξυπνες» τεχνικές για την ανάλυσή τους σε πραγματικό χρόνο. Η ομαδοποίηση (clustering) θεωρείται ιδιαίτερα σημαντική μέθοδος για την εξόρυξη ροών δεδομένων (data stream mining). Τα δεδομένα ομαδοποιούνται σύμφωνα με τις ομοιότητές τους, οι οποίες καθορίζονται με βάση την απόσταση ή την πυκνότητα. Έχουν προταθεί πολλές τεχνικές, αλλά στην προκειμένη περίπτωση προτιμούνται οι density-based μέθοδοι, οι οποίες έχουν την ικανότητα να σχηματίζουν clusters οποιασδήποτε μορφής εντοπίζοντας και τον θόρυβο. Οι density-based μέθοδοι ομαδοποίησης ροών δεδομένων αποτελούνται κυρίως από density grid-based μεθόδους [12] και density-based microclustering μεθόδους [13]. Στην πρώτη κατηγορία ο χώρος διασπάται σε έναν αριθμό density grids που σχηματίζουν μια δομή πλέγματος πάνω στο οποίο εκτελούνται όλες οι διαδικασίες ομαδοποίησης. Το κύριο πλεονέκτημα αυτής της μεθόδου είναι ο σύντομος χρόνος επεξεργασίας, ο οποίος είναι εξαρτώμενος από τον αριθμό των κελιών και όχι τον αριθμό των σημείων δεδομένων. Ενδέχεται, όμως, να έχουν χαμηλότερη ποιότητα και ακρίβεια. Στην δεύτερη κατηγορία, κρατιέται μία σύνοψη των clusters σε microclusters και σχηματίζουν τους τελικούς clusters από αυτούς. Προσφέρουν καλύτερη ποιότητα σε σχέση με τους grid-based, αλλά απαιτούν περισσότερο χρόνο υπολογισμού.

Για τον μετριασμό αυτών των προβλημάτων, στην παρούσα δημοσίευση προτείνεται μια υβριδική μέθοδος η οποία εκμεταλλεύεται τα πλεονεκτήματα και των δύο παραπάνω μεθόδων και ονομάζεται ***Hybrid density-based clustering for data stream (HDC-Stream)*** [14]. Η διαδικασία εκτελείται σε τρία βήματα τα οποία φαίνονται και στην εικόνα 5:

1. *Συγχώνευση ή χαρτογράφηση (Merging or mapping – MM-Step):*
   Κάθε νέο σημείο δεδομένων που διαβάζεται, συγχωνεύεται στον κοντινότερο minicluster (MIC) εάν η απόστασή του από το κέντρο του MIC είναι μικρότερη από r_{MIC}, αλλιώς αντιστοιχίζεται στο πλέγμα. (Ο minicluster είναι παρόμοια έννοια με τον microcluster το οποίο σχηματίζεται από ένα κελί του πλέγματος).
2. *Περικοπή πλέγματος και των miniclusters (Pruning grids and miniclusters – PGM-Step):*

Σε κάθε MIC, αν δεν προστεθεί καινούριο σημείο, το βάρος του θα φθίνει σταδιακά. Κάποια πλέγματα αν δε λάβουν νέα σημεία δεδομένων για μεγάλο χρονικό διάστημα, απομονώνονται και πρέπει να αφαιρούνται. Γι' αυτό, τα βάρη των κελιών και των miniclusters ελέγχονται περιοδικά κατά τον χρόνο περικοπής και εκείνα με βάρος μικρότερο από ένα κατώφλι απορρίπτονται, ελευθερώνοντας το χώρο μνήμης.

3. *Σχηματισμός τελικών ομάδων (Forming final clusters – FFC-Step):*
   Οι τελικοί clusters σχηματίζονται με βάση τους miniclusters που υπέστησαν περικοπή. Κάθε minicluster θεωρείται ως ένα εικονικό σημείο τοποθετημένο στο κέντρο του MIC με βάρος w_{MIC}. Υιοθετώντας την ιδέα της density connectivity από το [9], σχηματίζονται οι τελικοί clusters κάνοντας χρήση του modified DBSCAN αλγορίθμου.



5. HDC-Stream αλγόριθμος

Ο HDC-Stream αξιολογήθηκε σε σχέση με δύο άλλες γνωστές μεθόδους, τους αλγορίθμους DenStream και D-Stream. Για την αξιολόγηση της ποιότητας υιοθετήθηκαν δύο δημοφιλή μέτρα (measures), η καθαρότητα (purity) και η κανονικοποιημένη αμοιβαία πληροφορία (normalized mutual information - NMI). Στο πείραμα με τα συνθετικά δεδομένα (synthetic data), ο HDC-Stream παρουσίασε καθαρότητα ομαδοποίησης 98%, ενώ οι DenStream και D-Stream 82% και 78%, αντίστοιχα. Η ανωτερότητα του HDC-Stream επαληθεύεται και από τα αποτελέσματα του NMI. Το ίδιο αποδεικνύεται και στα πειράματα με τα πραγματικά δεδομένα, στα οποία η καθαρότητα του HDC-Stream φτάνει το 95%, ενώ των DenStream και D-Stream 86% και 76%, αντίστοιχα. Ο προτεινόμενος αλγόριθμος έχει επίσης συντομότερο χρόνο εκτέλεσης σε σύγκριση με τους άλλους δύο. Η ελάττωση της χρονικής πολυπλοκότητας επετεύχθη με τη

χρήση της grid-based ομαδοποίησης, η οποία μείωσε την χρονική πολυπλοκότητα της συγχώνευσης από $O(mi)$ σε $O(1)$ που είναι ο χρόνος χαρτογράφησης (mapping time). Συνοπτικά, ο HDC-Stream, κατάφερε να βελτιώσει την ποιότητα των αποτελεσμάτων της ομαδοποίησης, καθώς και να μειώσει τον χρόνο υπολογισμού. Παρόλα αυτά δεν κρίνεται κατάλληλος για χρήση σε κατανεμημένα περιβάλλοντα.

4.2.2 A Hybrid Approach to Clustering in Big Data

Στο [15] προτείνεται ένας ***νέος clusiVAT*** αλγόριθμος που αποτελεί επέκταση των ιδεών που παρουσιάστηκαν στο [16]. Με βάση την παλαιότερη δουλειά, ο clusiVAT παράγει single linkage (SL) clusters σε compact-separated (CS) δεδομένα. Ο νέος clusiVAT βασίζεται στις reordered dissimilarity images (RDIs) στις οποίες ζεύγη πληροφοριών ανομοιότητας (dissimilarity information) για ένα σύνολο δεδομένων n δειγματοληπτημένων αντικειμένων αναπαρίσταται ως μια εικόνα $n \times n$. Οι RDIs δημιουργούνται χρησιμοποιώντας τον Visual Assessment of Cluster Tendency (VAT) ο οποίος χρησιμοποιώντας έναν τροποποιημένο αλγόριθμο του Prim, ξανατακτοποιεί την input distance matrix και επισημαίνει τους πιθανούς clusters ως ένα σύνολο σκουρόχρωμων μπλοκ στη διαγώνιο της εικόνας. Έτσι, ο αριθμός των clusters μπορεί εύκολα να εκτιμηθεί οπτικά. Τα δείγματα ομαδοποιούνται με τη χρήση ενός συγγενούς του single linkage (SL) και ύστερα επεκτείνει μη επαναληπτικά τις ετικέτες στο υπόλοιπο σύνολο δεδομένων κάνοντας χρήση του nearest (object) prototype rule (NPR).

Η απόδοση του *νέου clusiVAT* αλγορίθμου συγκρίθηκε με αυτή των k-means, single pass k-means (spkm), online k-means (okm) και clustering using representatives (CURE). Για να αποδειχτεί η χρησιμότητα του clusiVAT από την άποψη χρόνου εκτέλεσης (CPU time ή process time) και PA, όπου $PA = \#(Correctly\ labeled\ samples)/\#(Total\ samples)$, εκτελέστηκαν πειράματα σε 24 2-D συνθετικά δεδομένα, εννιά συνθετικά δεδομένα υψηλών διαστάσεων και δύο μεγάλα σύνολα δεδομένων πραγματικού κόσμου. Για compact separated (CS) σύνολα δεδομένων, ο νέος clusiVAT δίνει 100% ακρίβεια σε πολύ λιγότερο χρόνο απ' ό,τι ο k-means και οι παραλλαγές του, καθώς και ο CURE. Για τα δισδιάστατα (2-D) μη-CS σύνολα δεδομένων, ο clusiVAT δίνει υψηλή ακρίβεια (≥99.8%) σε 12-18 φορές συντομότερο χρόνο εκτέλεσης από τον k-means και τις παραλλαγές του και 60-90 φορές συντομότερο από τον CURE. Για να φανεί η χρησιμότητα του αλγορίθμου για δεδομένα χωρίς ετικέτα, έγιναν πειράματα και σε δεδομένα χρήσης ενέργειας που συλλέχθηκαν από πραγματικό ασύρματο δίκτυο αισθητήρων (wireless sensor network – WSN) σε εσωτερικό χώρο γραφείων και αποδείχτηκε πως ο clusiVAT παρήγαγε clusters με separation index ή Dunn index (DI) πολύ μεγαλύτερο του 1. Επίσης, εφαρμόζοντας το Friedman test στα αποτελέσματα PA και DI για όλα τα σύνολα δεδομένων, προέκυψε

η κατάταξη απόδοσης των αλγορίθμων. Ξεκινώντας από την καλύτερη προς τη χειρότερη απόδοση, η κατάταξη (με τους αντίστοιχους λόγους PA/DI) είναι η εξής: clusiVAT (1.56), spkm (2.17), CURE (2.73), k-means (4.18) και okm (4.36).

4.2.3 A new credibilistic clustering algorithm

Ένας βασικός τύπος κατηγοριοποίησης των αλγορίθμων ομαδοποίησης τους κατατάσσει σε δύο κατηγορίες: crisp (σαφείς) και fuzzy (ασαφείς). Ένα από τα πιο γνωστά μοντέλα fuzzy ομαδοποίησης είναι ο *Fuzzy C-Means (FCM)* [17]. Αυτό το μοντέλο επιβάλλει το άθροισμα των βαθμών συμμετοχής (degree of membership) κάθε δεδομένου στους clusters να είναι ίσο με 1. Αν και πρόκειται για πολύ χρήσιμη μέθοδο, οι συμμετοχές (memberships) δεν αντιστοιχούν πάντα καλά στον "degree of belonging" των δεδομένων και μπορεί να είναι ανακριβής σε περιβάλλοντα με θόρυβο. Για την αντιμετώπιση αυτής της αδυναμίας του FCM και την απόκτηση συμμετοχών (memberships) που εξηγούν καλά τους degrees of belonging των δεδομένων, παρουσιάστηκε ο *Possibilistic C-Means (PCM)* [18], ο οποίος υπολογίζει έναν πιθανοτικό (possibilistic) τύπο συνάρτησης συμμετοχής (membership function) για να βρει τον degree of belonging. Το πρόβλημα του PCM είναι οι συμπίπτοντες (coincident) clusters, γιατί χαλαρώνοντας τον περιορισμό του FCM, παίρνει τον βαθμό συμμετοχής κάθε δεδομένου σε κάθε cluster, λαμβάνοντας υπόψη μόνο εκείνον τον cluster. Οπότε, επιτρέπει κάθε δεδομένο να ανήκει σε κάθε cluster ανεξάρτητα από τα άλλα δεδομένα και τους άλλους clusters. Σε αντίθεση με το μέτρο πιθανότητας (possibility measure), το μέτρο αξιοπιστίας (credibility measure) είναι self-dual, δηλαδή η αξιοπιστία συμμετοχής (credibility of belonging) κάθε δεδομένου σε κάθε cluster συν την αξιοπιστία συμμετοχής του στους υπόλοιπους clusters είναι ίση με 1. (Το μέτρο αξιοπιστίας παίρνει τιμές μεταξύ 0 και 1). Με αυτόν τον τρόπο, για την κατασκευή κάθε cluster λαμβάνεται υπόψη περισσότερη πληροφορία σχετικά με τους υπόλοιπους clusters.

Στην παρούσα δημοσίευση παρουσιάζεται ένα μοντέλο ομαδοποίησης με βάση την αξιοπιστία (credibilistic model), το ***Credibilistic Clustering Model (CCM)*** [19], το οποίο αξιοποιεί τα θετικά χαρακτηριστικά και των δύο μεθόδων (PCM και FCM), ενώ εξαλείφει τις ατέλειές τους χρησιμοποιώντας το *μέτρο αξιοπιστίας (credibility measure)* [20], το οποίο ορίστηκε στα πλαίσια της *θεωρίας αξιοπιστίας (credibility theory)* [21], έναν κλάδο των μαθηματικών για τη μελέτη των ασαφών φαινομένων. Όπως ήδη ειπώθηκε, το μέτρο αξιοπιστίας είναι self-dual. Όταν εφαρμόζεται σε fuzzy ομαδοποίηση, η ιδιότητα της self-duality αναγκάζει τον αλγόριθμο να αξιοποιήσει τα οφέλη του FCM χωρίς την επιβολή του περιορισμού του. Έτσι, οι πληροφορίες για τα κέντρα και τους βαθμούς συμμετοχής των άλλων clusters,

συμβάλλουν στην κατασκευή των βαθμών συμμετοχής των υπόλοιπων clusters. Επίσης, αποτρέπει την δημιουργία συμπιπτόντων clusters, γεγονός το οποίο είναι η ατέλεια του PCM, χρησιμοποιώντας όμως και το πλεονέκτημά του που είναι η αντιμετώπιση του θορύβου. Μολονότι υπήρξε χαλάρωση του περιορισμού του FCM, η self-duality διατηρεί τα πλεονεκτήματα αυτού του περιορισμού. Η θεωρία αξιοπιστίας έχει ξαναεφαρμοστεί για ομαδοποίηση, όμως το μέτρο αξιοπιστίας και ο διαχωρισμός των clusters (cluster separation) χρησιμοποιούνται πρώτη φορά στην αντικειμενική συνάρτηση του PCM. Στο [22] παρουσιάστηκε πρώτη φορά ο Credibilistic Clustering Algorithm (CCA) ο οποίος χρησιμοποίησε στον FCM το μέτρο αξιοπιστίας αντί των βαθμών συμμετοχής, για την αφαίρεση του περιορισμού του FCM.

Το μοντέλο που αναπτύχθηκε αφαιρεί τα προβλήματα των FCM και PCM, αντικαθιστώντας στον PCM το μέτρο πιθανότητας με το μέτρο αξιοπιστίας για την απόκτηση του degree of belonging των δεδομένων σε έναν cluster, το οποίο είναι ένα ασαφές γεγονός. Χρησιμοποιώντας διάφορα σύνολα δεδομένων αποδείχτηκε πως ο PCM πράγματι κάποιες φορές παράγει επιπλέον clusters, ενώ το CCM μοντέλο είναι πιο αποδοτικό από τους PCM και FCM.

4.2.4 A New Mechanism for RFID Clustering and Identification

Μία από τις σημαντικότερες τεχνολογίες που χρησιμοποιείται για την υλοποίηση του Διαδικτύου των Πραγμάτων είναι η ταυτοποίηση μέσω ραδιοσυχνοτήτων (RFID – Radio Frequency Identification). Η RFID τεχνολογία έχει την ικανότητα να εξάγει αυτόματα και ασύρματα πληροφορίες από μικροηλεκτρονικές ετικέτες (tags – microchip με κεραία) που είναι κολλημένες πάνω σε αντικείμενα, μέσω ραδιοκυμάτων. Ένα σημαντικό μειονέκτημα των RFID συστημάτων είναι η χαμηλή αποδοτικότητα στην αναγνώριση ετικετών λόγω του προβλήματος της περιπλοκής/σύγκρουσης (collision) των ετικετών. Αυτό συμβαίνει όταν πολλαπλές ετικέτες ενεργοποιούνται από τον αναγνώστη (reader) RFID ετικετών ταυτόχρονα και αντίστοιχα εκείνες αντανακλούν τα σήματά τους στον αναγνώστη την ίδια στιγμή – γεγονός που συγχέει τον αναγνώστη. Το πρόβλημα εντείνεται όταν πρόκειται για μεγάλο όγκο ετικετών μέσα στο ίδιο πεδίο ραδιοσυχνοτήτων, επομένως στα πλαίσια του ΔτΠ αποτελεί πρόκληση η μεγιστοποίηση του αριθμού των ετικετών που αναγνωρίζονται, ενώ ελαχιστοποιείται ο χρόνος που απαιτείται για αυτό – χωρίς καθυστερήσεις λόγω collisions.

Ο προτεινόμενος μηχανισμός [23] χρησιμοποιεί δύο αναγνώστες – αντί του ενός – που έχουν την ίδια εμβέλεια «ερωταπάντησης»/αναζήτησης (interrogation) και ομαδοποιεί τις ετικέτες σε αυτή τη ζώνη. Ο μηχανισμός αποτελείται από τα εξής βήματα:

1. Ο αναγνώστης (reader) 1 παίζει το ρόλο του «επικεφαλής του cluster» (cluster-head) και συλλέγει τις ταυτότητες (IDs) από τις ετικέτες που βρίσκονται στη ζώνη αναζήτησής του (interrogation zone). Αυτές οι ετικέτες ομαδοποιούνται σε clusters ώστε να ελαχιστοποιηθεί το ρίσκο «σύγκρουσης» (collision). Για αυτή την ομαδοποίηση των ετικετών χρησιμοποιείται ο Progressive Scanning (PS) Algorithm [24], ο οποίος βελτιώνει την απόδοση της συλλογής δεδομένων, ομαδοποιώντας τις ετικέτες ανάλογα με τα επίπεδα ενέργειας – δηλαδή ο αναγνώστης ξεκινά τη μετάδοση (transmission) με το ελάχιστο επίπεδο ενέργειας, συνεχίζει σταδιακά έως το μέγιστο επιτρεπτό και έτσι αναγνωρίζει τις ετικέτες ανά ομάδες.
2. Αφού εκτιμήσει τον αριθμό των ετικετών στην ζώνη αναζήτησής του και εφόσον η διακίνηση (throughput = ο λόγος των αναμενόμενων τιμών επιτυχημένων χρονοθυρίδων/ συνολικό αριθμό χρονοθυρίδων) του συστήματος είναι μέγιστη όταν ο αριθμός ετικετών είναι περίπου ίδιος με τον αριθμό των χρονοθυρίδων (time slots), προσαρμόζει το μέγεθος πλαισίου (frame size) κατάλληλα.
3. Ο αναγνώστης 1 ξεκινά να αναγνωρίζει τις ετικέτες ανά cluster. Εάν προκύψει κάποια περιπλοκή (collision) ετικετών, το θέμα λύνεται χρησιμοποιώντας κάποιον αλγόριθμο από την υπάρχουσα βιβλιογραφία. Έτσι, στο τέλος αυτού του βήματος, ο αναγνώστης 1 θα έχει συγκεντρώσει όλες τις ταυτότητες (IDs) των ετικετών από τους διάφορους clusters που βρίσκονται εντός του πεδίου αναζήτησής του.
4. Ο αναγνώστης 2 λαμβάνει μέσω κάποιου ασύρματου δικτύου, όπως το ZigBee, τις συγκεντρωμένες ταυτότητες από τον αναγνώστη 1.
5. Αφού έχει λάβει όλες τις ταυτότητες από τον αναγνώστη 1, στέλνει αιτήματα αναγνώρισης στις ετικέτες, οι οποίες με τη σειρά τους αποκρίνονται στέλνοντας τα δεδομένα που είναι αποθηκευμένα στη μνήμη τους.

Με αυτόν τον τρόπο, ο αναγνώστης 2 λαμβάνει αυτόματα τα δεδομένα από τις ετικέτες, χωρίς να χάνει χρόνο επιλύοντας προβλήματα σύγκρουσης (collision) – διαδικασία που αναλαμβάνει ο αναγνώστης 1.

Ο μηχανισμός αυτός συγκρίθηκε μέσω προσομοιώσεων με τους *Dynamic Framed Slotted Aloha (DFSA)* [25] και *Enhanced Dynamic Framed Slotted Aloha (EDFSA)* [26] αλγορίθμους. Ο *DFSA* στοχεύει στην βελτίωση της απόδοσης του συστήματος αναγνώρισης, αλλάζοντας το μέγεθος πλαισίου σύμφωνα με τον αριθμό των ετικετών. Το μέγεθος του νέου πλαισίου καθορίζεται με βάση πληροφορίες από το προηγούμενο. Μετά το πρώτο πλαίσιο, ο αναγνώστης αποφασίζει εάν θα αυξήσει, μειώσει, ή κρατήσει

το ίδιο μέγεθος πλαισίου. Αποδείχτηκε ότι αυτός ο αλγόριθμος είχε καλή απόδοση εάν το μέγεθος πλαισίου ήταν ίσο με τον αριθμό των ετικετών στην περιοχή αναγνώρισης του αναγνώστη. Ο *EDFSA* είναι μια μέθοδος ομαδοποίησης ετικετών, στην οποία ο αναγνώστης εκτιμά τον αριθμό τους και αν υπερβαίνει ένα κατώφλι (το μέγιστο μέγεθος πλαισίου), χωρίζει σε ομάδες τις ετικέτες και διαβάζει την κάθε ομάδα χωριστά. Αυτό βοηθάει στη βελτίωση της απόδοσης, αφού όταν ο αριθμός των ετικετών γίνεται μεγαλύτερος από το μέγεθος πλαισίου αυξάνεται η πιθανότητα για σύγκρουση. Με αυτόν τον τρόπο, αυξάνει την απόδοση κατά 85% σε σχέση με τον DFSA.

Η σύγκριση των παραπάνω αλγορίθμων με τον προτεινόμενο έγινε ως προς τον απαιτούμενο αριθμό χρονοθυρίδων (time slots) για την αναγνώριση ενός συνόλου ετικετών, και αποδείχτηκε ότι μπορεί να επιτύχει καλύτερη επίδοση από τους άλλους δύο. Συνοπτικά, ο προτεινόμενος μηχανισμός αύξησε την αποδοτικότητα της διαδικασίας αναγνώρισης, μειώνοντας τον απαιτούμενο χρόνο για να αναγνωριστούν όλες οι ετικέτες, χάρη στη διάταξη με τους δύο αναγνώστες. Επίσης, μειώνει τον αριθμό των συγκρουόμενων ετικετών (collided tags) χάρη στην ομαδοποίηση των ετικετών με τον PS αλγόριθμο.

4.2.5 An efficient and scalable density-based clustering algorithm for datasets with complex structures

Ο Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9] είναι από τους πιο ευρέως χρησιμοποιημένους αλγορίθμους ομαδοποίησης για χωρικά σύνολα δεδομένων. Μπορεί να ανιχνεύσει clusters οποιουδήποτε σχήματος και μεγέθους, να αναγνωρίσει αυτόματα σημεία θορύβου και να προσδιορίσει τον αριθμό των clusters χωρίς τη συμβολή του χρήστη. Ωστόσο, έχει κάποιους προβληματικούς περιορισμούς:

1. Η απόδοσή του εξαρτάται από δύο παραμέτρους, τις ε και $MinPts$, όπου το ε συμβολίζει την μέγιστη ακτίνα της γειτονιάς από το σημείο παρατήρησης και το $MinPts$ τον ελάχιστον αριθμό σημείων δεδομένων (data points) που περιέχονται σε μια τέτοια γειτονιά. Οι κατάλληλες παράμετροι για διάφορα σύνολα δεδομένων χωρίς κάποια άλλη πληροφορία, είναι δύσκολο να εκτιμηθούν.
2. Η κατανάλωση χρόνου για την εύρεση των πλησιέστερων γειτόνων κάθε αντικειμένου είναι χρονοβόρα και στέκεται εμπόδιο στην επεκτασιμότητα του αλγορίθμου.
3. Είναι ευαίσθητος στη σειρά με την οποία εισάγονται τα δεδομένα, κατά συνέπεια καταλήγει σε διαφορετικά αποτελέσματα ανάλογα με την σειρά.

4. Ο DBSCAN δεν έχει την ικανότητα να αναγνωρίσει παρακείμενους clusters με διαφορετικές πυκνότητες.

Επίσης, συχνά αγνοείται η ανίχνευση των οριακών σημείων (border points).

Στην παρούσα δημοσίευση [10] επιλύονται επιτυχώς τα παραπάνω προβλήματα. Πρώτον, βελτιώθηκε η *traditional locality sensitive hashing* μέθοδος για την γρήγορη αναζήτηση των πλησιέστερων γειτόνων. Δεύτερον, επαναπροσδιορίζεται το *influence space* κάθε αντικειμένου, και πλέον λαμβάνει υπόψη όχι μόνο τους πλησιέστερους γείτονες, αλλά και τους αντίστροφα πλησιέστερους γείτονες. Μια παλαιότερη βελτίωση του DBSCAN είναι ο IS-DBSCAN [27], ο οποίος χρησιμοποιεί επίσης το influence space. Η διαφορά του προτεινόμενου αλγορίθμου, όμως, είναι πως επαναπροσδιορίζει την ιδέα του *neighborhood relationship*, προτείνει την ιδέα του *core density reachable* και εισάγει μια νέα μέθοδο για την αναζήτηση των γειτόνων. Το πλεονέκτημα του IS-DBSCAN στην αναγνώριση παρακείμενων clusters διαφορετικών πυκνοτήτων διατηρείται, ενώ παράλληλα ξεπερνιέται η δυσκολία ανίχνευσης οριακών σημείων και επιταχύνεται η αναζήτηση των πλησιέστερων γειτόνων.

Στη συνέχεια, αναλύονται οι τρεις αυτές μέθοδοι οι οποίες ενσωματώνονται στον DBSCAN και αποτελούν πλέον τον προτεινόμενο ***DBSCAN based on Influence Space and Detecting of border points (ISB-DBSCAN)*** [10] αλγόριθμο.

Improved p-stable locality sensitive hashing (RLSH) method

Η βελτιωμένη p-stable LSH (RLSH) μέθοδος βασίζεται στην *p-stable locality sensitive hashing* μέθοδο [28], η οποία ψάχνει τους πλησιέστερους γείτονες κατά προσέγγιση, με στόχο την επιτάχυνση του ερωτήματος (query) αυτού του βήματος. Η RLSH βελτιώνει τον p-stable LSH αλγόριθμο με σκοπό τη μείωση του απαιτούμενου χώρου για την αποθήκευση των πινάκων κατακερματισμού (hash tables) και της κατανάλωσης χρόνου.

Influence space (IS)

Στην παρούσα δημοσίευση, χρησιμοποιείται η influence space (IS)-γειτονιά ή IS_k-γειτονιά, αντί της ε-γειτονιάς που χρησιμοποιεί ο DBSCAN. Το influence space αποτυπώθηκε πρώτα στο [27] για την εξόρυξη local outliers. Ο αριθμός IS_k κάθε αντικειμένου χρησιμοποιείται για να εκτιμηθεί εάν ένα σημείο είναι σημείο πυρήνα (core point) ή όχι. Σε αντίθεση με τον ορισμό της ε-γειτονιάς, ο ορισμός της IS_k-γειτονιάς στην παρούσα δημοσίευση, εκτός από την πλησιέστερη γειτονιά κάθε αντικειμένου, λαμβάνει υπόψη και την αντίστροφα πλησιέστερη γειτονιά. Η IS_k-γειτονιά εγγυάται την ευαισθησία του προτεινόμενου ISB-DBSCAN σε τοπικές αλλαγές της πυκνότητας, ώστε να μπορεί να εντοπίζει παρακείμενων clusters με

διαφορετικές πυκνότητες. Επιπλέον, λόγω της συμμετρίας του influence space, ένα άλλο δυνατό σημείο του προτεινόμενου αλγορίθμου είναι η ικανότητά του να επιλέγει τυχαία κάποιο αντικείμενο ως σημείο εκκίνησης για την επέκταση των clusters. Παράλληλα, μειώνει επιτυχώς τον αριθμό των παραμέτρων, απαιτώντας μόνο τον αριθμό των k-πλησιέστερων γειτόνων για την εύρεση του influence space IS_k.

Core density reachable

Στον DBSCAN, σημεία που είναι προσπελάσιμα με βάση την πυκνότητα (density reachable) από ένα σημείο p, τους ανατίθεται η ίδια ετικέτα (label) με του p. Στην αλυσίδα από density reachable σημεία που δημιουργείται, το τελευταίο είναι οριακό αντικείμενο (border object). Τα οριακά αντικείμενα δεν συμβάλλουν στον μηχανισμό επέκτασης αυτών των αλυσίδων. Γι' αυτό με σκοπό την αποσύνδεση των σημείων αυτών από την αλυσίδα, προτείνεται η νέα ιδέα που ονομάζεται *core density reachable*, η οποία βασίζεται στην ιδέα που έχει παρουσιαστεί στο [29]. Σε εκείνη τη δημοσίευση, ο ορισμός του core density reachable βασίζεται στη σχέση της ε-γειτονιάς, ενώ στην παρούσα, η core density reachable ορίζεται στην IS_k-γειτονιά. Τα οριακά σημεία που αναγνωρίζονται λαμβάνουν την ίδια ετικέτα με εκείνη που έχει το πλησιέστερο σημείο πυρήνα στα οριακά σημεία. Σημεία των οποίων τα influence spaces δεν έχουν σημεία ή έχουν μόνο σημεία θορύβου (noisy points) θεωρούνται σημεία θορύβου. Με αυτόν τον μηχανισμό, μειώνεται η πιθανότητα ένα οριακό σημείο να θεωρηθεί λανθασμένα ως σημείο θορύβου.

Ο ISB-DBSCAN, συγκρίθηκε με τον DBSCAN και τον IS-DBSCAN [27] ως προς τρία σημεία: τον χρόνο αναζήτησης των πλησιέστερων γειτονιών, την ρύθμιση των παραμέτρων και την απόδοση στην ανίχνευση οριακών σημείων. Στα πειράματα, ο ISB-DBSCAN χρειάστηκε λιγότερο χρόνο αναζήτησης από τον p-stable LSH στην περίπτωση που η βάση δεδομένων είναι μεγάλου μεγέθους (>36.000 σημεία). Ως προς τη ρύθμιση παραμέτρων, ο προτεινόμενος αλγόριθμος, όχι μόνο μειώνει τον αριθμό των απαιτούμενων παραμέτρων, αλλά διευρύνει και το πεδίο των τιμών των παραμέτρων που πετυχαίνουν επιδόσεις υψηλής ποιότητας. Σε σχέση με τον DBSCAN, το συμπέρασμα που προέκυψε είναι πως ο ISB-DBSCAN έχει μεγαλύτερη πιθανότητα να πετύχει καλύτερο αποτέλεσμα. Όσον αφορά την ανίχνευση οριακών σημείων, τόσο ο DBSCAN όσο και ο ISB-DBSCAN μπορούν να ανιχνεύσουν αυτά, αλλά και τα σημεία θορύβου, ενώ ο IS-DBSCAN αποτυγχάνει στη διάκρισή τους. Αυτό συμβαίνει για το λόγο ότι σημεία που είναι density reachable από το σημείο παρατήρησης, ο αριθμός των οποίων είναι μικρότερος από την παράμετρο k, θεωρούνται ως σημεία θορύβου από τον IS-DBSCAN. Ο ISB-DBSCAN έλυσε αυτό το πρόβλημα φιλτράροντας όλα τα πιθανά οριακά σημεία και σημεία θορύβου σε κάθε βήμα επέκτασης των clusters. Συνοψίζοντας, ο ISB-DBSCAN είναι απόλυτα καλύτερος όταν χρησιμοποιείται σε σύνολα

δεδομένων πολλών διαστάσεων και σημείων, επομένως μπορεί να εφαρμοστεί σε σύνολα δεδομένων με πολυπλοκότερη δομή.

4.2.6 Bias-correction fuzzy clustering algorithms

Οι ασαφείς (fuzzy) αλγόριθμοι ομαδοποίησης αποτελούν επέκταση της σαφούς ομαδοποίησης (hard clustering) και βασίζονται σε χωρίσματα ασαφούς συμμετοχής (fuzzy membership partitions). Ο Fuzzy C-Means (FCM) είναι ο πιο συχνά χρησιμοποιούμενος fuzzy αλγόριθμος. Παρόλα αυτά, ο FCM και οι γενικεύσεις του όπως ο Generalized FCM (GFCM) [30] συνήθως επηρεάζονται από τις αρχικοποιήσεις (initializations), δηλαδή μπορεί να επιστρέψουν μη ικανοποιητικά αποτελέσματα ομαδοποίησης όταν χρησιμοποιούνται κακές αρχικοποιήσεις. Στο παρόν άρθρο [31] προτείνεται ένας όρος (term) για διόρθωση του σφάλματος μεροληψίας (bias-correction term) με μια εξίσωση ενημέρωσης (updating equation) για την διόρθωση των συνεπειών των αρχικοποιήσεων.

Ο GFCM μετατρέπεται σε bias-correction GFCM ως εξής: Αρχικά, θεωρείται μια πιθανότητα μάζας (probability mass) p_i για κάθε κέντρο cluster $a_i, i = 1, \dots, c$ με $\sum_{i=1}^{c} p_i = 1$. Η probability mass p_i μπορεί να χρησιμοποιηθεί για την αναπαράσταση του ποσοστού του κέντρου ενός cluster a_i στους c clusters. Θεωρητικά, ο όρος $-\ln(p_i)$ μπορεί να αναπαραστήσει την πληροφορία σχετικά με την εμφάνιση του κέντρου ενός cluster, a_i. Επομένως, η συνολική πληροφορία με βάση τα ασαφή c-χωρίσματα (fuzzy c-partitions) μ_{ik} μπορεί να εκφραστεί ως $-\sum_{k=1}^{n}\sum_{i=1}^{c} \mu_{ik}^{m} \ln(p_i)$ το οποίο ονομάζεται και εντροπία (entropy). Για την απόκτηση περισσότερων πληροφοριών μπορεί να υπολογιστεί το βέλτιστο p_i ελαχιστοποιώντας την entropy. Τέλος, ο bias correction όρος $-w\sum_{k=1}^{n}\sum_{i=1}^{c} \mu_{ik}^{m} \ln(p_i)$, όπου $w^{(t)} = (0.99)^t$ με t τον αριθμό των επαναλήψεων, προσαρμόζεται στην αντικειμενική συνάρτηση του GFCM. Με παρόμοιο τρόπο αναπτύχθηκαν και οι τρεις τύποι bias-correction GFCM αλγορίθμων, οι οποίοι είναι ο **BFCM**, ο **bias-correction GK (BGK)** και ο **bias-correction ICS (BICS)**.

Οι προτεινόμενοι BFCM, BGK και BICS αλγόριθμοι [31] συγκρίθηκαν μέσω πειραμάτων με τους FCM, Gustafson and Kessel (GK) και Inter-Cluster Separation (ICS). Η αποτελεσματικότητα και ανωτερότητα των προτεινόμενων αλγορίθμων υποδεικνύεται από τον αριθμό των βέλτιστων αποτελεσμάτων ομαδοποίησης, τα ποσοστά σφάλματος και τη ρίζα των μέσως τετραγωνικών σφαλμάτων (root mean squared errors – RMSEs) που χρησιμοποιήθηκαν ως κριτήρια αξιολόγησης. Εξετάστηκαν αρκετές περιπτώσεις και δόθηκαν παραδείγματα τα οποία είχαν ως συμπέρασμα πως σε γενικές γραμμές, αλλά όχι πάντα, οι bias-correction εκδοχές των αλγορίθμων είχαν καλύτερη απόδοση. Ενδεικτικά, το μέσο

ποσοστό σφάλματος των FCM, BFCM, GK και BGK ήταν για το σύνολο των banana-shaped δεδομένων 10.78%, 12.34%, 4.38% και 0%, αντίστοιχα. Είναι φανερό πως ο BGK είναι ο πιο αποτελεσματικός στην ανάλυση αυτού του τύπου δεδομένων, όμως, η αποτελεσματικότητα του BFCM δεν είναι υψηλότερη από του απλού FCM.

4.2.7 Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm

Ο Fuzzy c-means (FCM) αλγόριθμος είναι μια συχνά χρησιμοποιούμενη μέθοδος στην αναγνώριση προτύπων (pattern recognition). Έχει το πλεονέκτημα ότι δίνει καλά αποτελέσματα μοντελοποίησης, αν και δεν είναι ικανός να προσδιορίσει τον αριθμό των clusters από μόνος του. Ο FCM βασισμένος στην αντικειμενική συνάρτηση (objective function) εφαρμόζεται ευρέως λόγω της ισχυρής του ικανότητας στην τοπική αναζήτηση (local search) και της γρήγορης ταχύτητας σύγκλισης. Έχει, όμως, κάποια ελαττώματα: είναι ευαίσθητος στον θόρυβο και στα απομονωμένα δεδομένα, όπως και στο αρχικό κέντρο ομαδοποίησης (initial clustering center) και συγκλίνει εύκολα σε ένα τοπικό ακρότατο. Στοχεύοντας στα προβλήματα του FCM, προτείνεται ένας kernel-based FCM για την βελτιστοποίηση του FCM, ο οποίος ονομάζεται ***GAKFCM*** [32] και είναι συνδυασμός ενός βελτιωμένου γενετικού αλγορίθμου και του Kernel-based fuzzy c-means (KFCM). Πρώτα χρησιμοποιείται ο βελτιωμένος γενετικός αλγόριθμος για να βελτιστοποιήσει το αρχικό κέντρο ομαδοποίησης και ύστερα ο KFCM για να βελτιώσει την απόδοση του FCM.

Τα αποτελέσματα έδειξαν ότι ο προτεινόμενος GAKFCM αλγόριθμος εξαλείφει αποτελεσματικά τα ελαττώματα του FCM, είναι ακριβής και βελτιώνει σε μεγάλο βαθμό την απόδοση.

4.2.8 An Adaptive Meta-Heuristic Search for the Internet of Things

Ο προτεινόμενος ***AntClust*** [33] είναι ένας μετα-ευρετικός αλγόριθμος εμπνευσμένος από τον ant clustering αλγόριθμο και πρόκειται για μια context-aware μέθοδο για την ομαδοποίηση αισθητήρων υπό τη μορφή Σημασιολογικών Δικτύων Επικάλυψης Αισθητήρων (Sensor Semantic Overlay Networks – SSONs), στα οποία αισθητήρες με παρόμοια συμφραζόμενα συγκεντρώνονται σε κοινό cluster. Έτσι, δίνεται η δυνατότητα αναζήτησης και επιλογής των πιο σχετικών αισθητήρων με βάση το ενδιαφέρον του χρήστη. Η έλλειψη της χρήσης ευρετικών αλγορίθμων για την αναζήτηση αισθητήρων ήταν αυτή που

ώθησε τους ερευνητές στην ανάπτυξη του AntClust με σκοπό τη μείωση του χώρου αναζήτησης και κατά συνέπεια την βελτίωση της απόδοσης της context-aware αναζήτησης αισθητήρων στο IoT.

Αρχικά, οι αισθητήρες ομαδοποιούνται με βάση το είδος τους (πχ. αισθητήρες που παρακολουθούν την κίνηση, τον καιρό, κλπ.) για τη δημιουργία SSONs. Ύστερα, εκτελείται ο AntClust για να ομαδοποιήσει τους αισθητήρες με βάση τις ιδιότητες των συμφραζομένων (πχ. ακρίβεια, αξιοπιστία, διαθεσιμότητα, ενέργεια, κόστος, κλπ.). Επιπλέον, εφαρμόζονται κάποιες προσαρμογές για τη μείωση του κόστους της διαδικασίας εύρεσης αισθητήρων και προτείνεται μια προσαρμοστική στρατηγική για την διατήρηση της απόδοσης ενάντια στην δυναμικότητα (dynamicity) που περιβάλλοντος του IoT.

Αρχικά στον αλγόριθμο οι αισθητήρες διασκορπίζονται τυχαία σε ένα πλέγμα έτσι ώστε κάθε κελί να περιέχει το πολύ έναν αισθητήρα. Το πλέγμα προσομοιώνεται με έναν δισδιάστατο πίνακα, στον οποίο κάθε στοιχείο περιλαμβάνει έναν αριθμό αισθητήρων. Ύστερα, διανέμονται τα «μυρμήγκια» στο πλέγμα και ξεκινούν να κινούνται. Κάθε μυρμήγκι υπολογίζει την *συνάρτηση πιθανότητας παραλαβής ενός αισθητήρα (pickup probability function)* για τους αισθητήρες που συναντά στην πορεία του. Εάν είναι δυνατό να παραλάβει έναν αισθητήρα, τότε θα το κάνει. Το μυρμήγκι συνεχίζει την τυχαία κίνησή του στο πλέγμα και υπολογίζει την *συνάρτηση πιθανότητας παράδοσης ενός αισθητήρα (drop probability function)* για τα κενά κελιά στην πορεία του. Αν σε κάποιο κενό κελί υπάρχει η πιθανότητα παράδοσης (drop probability) τότε το μυρμήγκι «ρίχνει» τον αισθητήρα και συνεχίζει την πορεία του για να παραλάβει έναν άλλον αισθητήρα. Ο υπολογισμός των συναρτήσεων πιθανότητας παραλαβής και παράδοσης βασίζεται στην ομοιότητα μεταξύ του επιθυμητού αισθητήρα και των γειτονικών αισθητήρων. Προκειμένου να αποφευχθεί η σπατάλη χρόνου στην διαδικασία αναζήτησης, τα μυρμήγκια προσεγγίζουν άλλους αισθητήρες αμέσως μετά την απελευθέρωση του προηγούμενου. Η έκταση των ομοιοτήτων μεταξύ αισθητήρων και των γειτονικών αισθητήρων κατά την διαδικασία παραλαβής/παράδοσης (picking up/dropping procedure) υπολογίζεται με μια συνάρτηση $f(sn_i)$ της οποίας η ιδέα πηγάζει από εκείνη που παρουσιάστηκε στο [34]. Για τον υπολογισμό της απόστασης μεταξύ δύο αισθητήρων χρησιμοποιείται η Ευκλείδεια απόσταση στον πολυδιάστατο χώρο. Ο μέσος όρος του συνόλου των αποστάσεων μεταξύ αισθητήρων υπολογίζεται με βάση τον τύπο που παρουσιάστηκε στο [35]. Αφού εκτιμηθούν οι ομοιότητες, υπολογίζονται οι pick up και drop πιθανότητες με την κύρια ιδέα των τύπων να σχετίζεται με το [36].

Για την προσαρμογή του ant clustering αλγορίθμου στο πρόβλημα ομαδοποίησης αισθητήρων, θεωρείται ότι κάθε μυρμήγκι μπορεί να απομνημονεύσει τον αριθμό των clusters (CN) που βρίσκεται ο αισθητήρας (sn_i). Η short-term memory προτάθηκε στο [34] και δίνει τη δυνατότητα στο μυρμήγκι να

θυμάται κάποιες από τις τοποθεσίες όπου έχει επιτυχώς παραδώσει έναν αισθητήρα. Αρχικά, όλοι οι αισθητήρες τοποθετούνται σε μια λίστα που ονομάζεται UnvisitedSensors. Όσο υπάρχουν αισθητήρες σε αυτή τη λίστα, κάθε μυρμήγκι χωρίς φορτίο φορτώνεται με έναν τυχαία επιλεγμένο αισθητήρα από τη λίστα, χωρίς τον υπολογισμό της picking up πιθανότητάς του και ύστερα διαγράφεται απ' τη λίστα. Για κάθε μυρμήγκι που είναι φορτωμένο με έναν αισθητήρα, εντοπίζεται ο περισσότερο παρόμοιος αισθητήρας (most similar sensor – MSS) από την βραχυπρόθεσμη μνήμη του και προσπαθεί να ρίξει τον sn_i στον ίδιο cluster με του MSS. Εάν τα καταφέρει, ο CN του sn_i αλλάζει και γίνεται ο CN του MSS. Αλλιώς, το μυρμήγκι ρίχνει τον sn_i σε ένα τυχαίο κενό κελί του πλέγματος και ρυθμίζει τον CN του sn_i σε νέο νούμερο. Μετά το βήμα αρχικοποίησης ξεκινά το κυρίως μέρος της ομαδοποίησης. Οι βασικές διαφορές τους είναι πως 1) στον βήμα αρχικοποίησης αγνοείται η πιθανότητα παραλαβής ενός αισθητήρα και 2) στο κυρίως βήμα υπάρχει διαφορετική αντιμετώπιση της κατάστασης που το μυρμήγκι αποτυγχάνει να παραδώσει τον sn_i στον ίδιο cluster με τον MSS στην βραχυπρόθεσμη μνήμη. Σε αυτή την περίπτωση, το μυρμήγκι κινείται τυχαία πάνω στο πλέγμα και υπολογίζει την συνάρτηση πιθανότητας παράδοσης για τα κενά κελιά που συναντά στην πορεία του. Εάν η πιθανότητα παράδοσης ενός αισθητήρα σε ένα κενό κελί υπάρχει, το μυρμήγκι θα παραδώσει τον αισθητήρα, και το CN του sn_i θα ρυθμιστεί όπως το CN του MSS στην S^2 γειτονιά του μυρμηγκιού. Αλλιώς, ο sn_i «αφήνεται» σε ένα τυχαίο κενό κελί χωρίς αλλαγή του CN.

Διατήρηση της απόδοσης της ομαδοποίησης σε δυναμικά δίκτυα αισθητήρων: Στο πλαίσιο ενός δυναμικού IoT περιβάλλοντος, η προσαρμοστικότητα είναι βασικό χαρακτηριστικό των υπηρεσιών καθώς παρέχει τον τρόπο σε μια εφαρμογή να προσαρμόζεται σε νέες σχετικές με τα συμφαζόμενα απαιτήσεις. Προκειμένου να διατηρηθεί η απόδοση της ομαδοποίησης, οι clusters ενδέχεται να χρειάζονται μια τροποποίηση όταν το δίκτυο αισθητήρων αλλάζει, δηλαδή ύστερα από την άφιξη και αναχώρηση αισθητήρων. Όταν το SSON είναι αρκετά μεγάλο, μερικές αλλαγές θα έχουν μικρή επίδραση στη συνολική απόδοση. Γι' αυτό ορίστηκε ένα κατώφλι δ_c ώστε να επαναλαμβάνεται η διαδικασία ομαδοποίησης μόνο όταν ο αριθμός αλλαγών ξεπερνά το δ_c. Έτσι, ο αλγόριθμος Re-clustering Procedure ελέγχει τον αριθμό αλλαγών και πράττει ανάλογα: 1) Εάν ο αριθμός είναι μικρότερος από το κατώφλι, για κάθε νέο αισθητήρα επιλέγεται ο καταλληλότερος cluster-head και ο αισθητήρας γίνεται μέλος του αντίστοιχου cluster. Εάν αλλάξουν οι ιδιότητες των συμφραζομένων κάποιου αισθητήρα, επιλέγεται ο περισσότερο παρόμοιος με βάση τις καινούριες ιδιότητες cluster-head και ο αισθητήρας προστίθεται στον σχετικό cluster. Όσο για έναν αισθητήρα που αποχωρεί από το δίκτυο, απλώς αφαιρείται από τους clusters. 2) Εάν ο αριθμός των αλλαγών υπερβαίνει το κατώφλι, ο AntClust αλγόριθμος εκτελείται ξανά

για να αναδιοργανώσει το SSON – αυτή τη φορά όμως εκτελείται σε λιγότερο χρόνο αφού αρκετοί αισθητήρες παραμένουν στην ίδια θέση.

Από τους αλγορίθμους που αφορούν την context-aware αναζήτηση αισθητήρων, μόνο ο CASSARAM [37] λαμβάνει υπόψη ένα μεγάλο εύρος ιδιοτήτων συμφραζομένων των αισθητήρων στη διαδικασία αναζήτησης, γι' αυτό επιλέχθηκε για τη σύγκριση με τον AntClust. Ο χρόνος εκτέλεσης της προτεινόμενης μεθόδου προέκυψε σημαντικά λιγότερος από του CASSARAM, όμως η ακρίβεια των επιλεγμένων αισθητήρων είναι ελαφρώς μικρότερη. Σε γενικές γραμμές, η ομαδοποίηση των αισθητήρων στα SSONs αύξησε σημαντικά την απόδοση της context-aware αναζήτησης αισθητήρων στο IoT και οι ρυθμίσεις που εφαρμόστηκαν στον αλγόριθμο βελτίωσαν περαιτέρω την ποιότητα της ομαδοποίησης. Η προτεινόμενη μέθοδος αποδείχτηκε επίσης πως παρουσιάζει υψηλή επεκτασιμότητα.

4.2.9 An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering

Ο k-means είναι από τους πιο σημαντικούς διαμεριστικούς (partitional) αλγορίθμους. Ξεκινάει αρχικοποιώντας k κέντρα των clusters. Τα διανύσματα εισόδου (σημεία δεδομένων) αντιστοιχίζονται σε έναν από τους υπάρχοντες clusters με βάση το τετράγωνο της Ευκλείδειας απόστασης από τους clusters. Ύστερα υπολογίζεται το μέσο (κεντροειδές) του κάθε cluster ώστε να ενημερωθεί το κέντρο των clusters (cluster center). Αυτή η ενημέρωση προκύπτει ως αποτέλεσμα της αλλαγής της συμμετοχής (membership) του κάθε cluster. Οι διαδικασίες αντιστοίχισης των διανυσμάτων εισόδου και ενημέρωσης των κέντρων των clusters επαναλαμβάνονται μέχρι οι τιμές των κέντρων των clusters να μην μεταβάλλονται. Παρόλα αυτά, ο k-means έχει κάποια μειονεκτήματα. Η αντικειμενική του συνάρτηση δεν είναι κυρτή (convex) και μπορεί να περιέχει πολλά τοπικά ελάχιστα (local minima). Συνεπώς, κατά τη διαδικασία ελαχιστοποίησης της αντικειμενικής συνάρτησης, υπάρχει η πιθανότητα εγκλωβισμού σε κάποιο τοπικό ελάχιστο. Γι' αυτό, το αποτέλεσμα του k-means εξαρτάται σε μεγάλο βαθμό από την αρχική επιλογή των κέντρων των clusters.

Ο *Modified Imperialist Competitive Algorithm (MICA)* είναι μία από τις εξελικτικές (evolutionary) μεθόδους που αναπτύχθηκαν για την επίλυση δύσκολων προβλημάτων βελτιστοποίησης, με καλύτερη απόκριση και γρήγορη σύγκλιση σε σχέση με τις συνήθεις εξελικτικές μεθόδους. Η βασική ιδέα του MICA είναι ο διαχωρισμός χωρών (countries) σε δύο είδη: ιμπεριαλιστικά κράτη (imperialist states) και αποικίες (colonies). Ο ανταγωνισμός των ιμπεριαλιστικών κρατών και η τακτική αφομοίωσης

(assimilation policy) αναγκάζουν τις αποικίες να συγκλίνουν στη βέλτιστη θέση. Οι αρχικές χώρες δημιουργούνται με χρήση ενός εξελικτικού αλγορίθμου και ο κανόνας κίνησης (movement rule) του MICA εφαρμόζεται για την κατάρρευση αδύναμων αυτοκρατοριών (empires) τα οποία αναλαμβάνονται από ισχυρά empires. Αν και ο MICA λαμβάνεται υπόψη ως μια ισχυρή τεχνική, ενδέχεται να εγκλωβιστεί σε τοπικά βέλτιστα, ειδικά όταν ο αριθμός των ιμπεριαλιστικών κρατών αυξάνεται. Για την άμβλυνση αυτού του μειονεκτήματος, η μετάλλαξη (mutation) μπορεί να εκτρέψει την κίνηση των αποικιών προς σχετικά ιμπεριαλιστικά κράτη, σε νέες θέσεις. Επίσης, χρησιμοποιείται μια Χαοτική τοπική αναζήτηση (Chaotic Local Search – CLS) για να εξαλείψει το πρόβλημα της σύγκλισης σε τοπικά βέλτιστα.

Ο νέος αλγόριθμος που προτείνεται και ονομάζεται ***Hybrid k-MICA*** [38] είναι μια νέα υβριδική εξελικτική μέθοδος βελτιστοποίησης, η οποία συνδυάζει τα πλεονεκτήματα του k-means και του MICA περιορίζοντας τα μειονεκτήματά τους. Σκοπός είναι η βέλτιστη ομαδοποίηση n αντικειμένων σε k clusters. Ύστερα από την δημιουργία των αρχικών χωρών, εφαρμόζεται ο k-means για να βελτιώσει την θέση των αποικιών. Υπάρχουν διάφοροι τρόποι συνδυασμού των δύο αυτών μεθόδων, εκείνος όμως με την καλύτερη ταχύτητα σύγκλισης και ακρίβεια, βρέθηκε να είναι ο εξής: Ο πληθυσμός δημιουργείται με τον MICA και σχηματίζονται οι αρχικές αυτοκρατορίες, ύστερα εφαρμόζεται ο k-means για να βελτιώσει τη θέση των αποικιών των αυτοκρατοριών και των ιμπεριαλιστικών κρατών. Το αποτέλεσμα του αλγορίθμου αναλαμβάνει ξανά ο MICA. Η ακριβής λειτουργία του φαίνεται και στο παρακάτω διάγραμμα ροής της εικόνας 6. Αποτελείται από τα παρακάτω 14 βήματα:

1. *Παράγεται ένας τυχαίος αρχικός πληθυσμός.*
   Ένας αρχικός πληθυσμός δεδομένων εισόδου παράγεται μέσω αρχικοποίησης Chaos.
2. *Υπολογισμός της τιμής της αντικειμενικής συνάρτησης.*
   Υπολογίζεται η τιμή της αντικειμενικής συνάρτησης για κάθε χώρα.
3. *Ταξινόμηση του αρχικού πληθυσμού με βάση τις τιμές της αντικειμενικής συνάρτησης.*
4. *Επιλογή των ιμπεριαλιστικών κρατών.*
   Χώρες με την ελάχιστη αντικειμενική συνάρτηση επιλέγονται ως ιμπεριαλιστικά κράτη και οι υπόλοιπες σχηματίζουν τις αποικίες (colonies) αυτών των ιμπεριαλιστικών κρατών.
5. *Διαχωρισμός των αποικιών μεταξύ ιμπεριαλιστικών κρατών.*
   Οι αποικίες διαχωρίζονται με βάση την ισχύ κάθε ιμπεριαλιστικού κράτους.
6. *Εφαρμογή του k-means σε κάθε αυτοκρατορία.*
7. *Μετακίνηση των αποικιών προς τα ιμπεριαλιστικά τους κράτη, όπως στο βήμα 3.*
8. *Χρήση μετάλλαξης (mutation) για την αλλαγή κατεύθυνσης των αποικιών.*

(γίνεται αναφορά στον MICA)

9. *Έλεγχος του κόστους όλων των αποικιών σε κάθε αυτοκρατορία.*

   Κατά τη διάρκεια των προηγούμενων βημάτων, το κόστος κάθε αποικίας ενδέχεται να έχει αλλάξει. Επομένως, ελέγχονται τα κόστη όλων των αποικιών μιας αυτοκρατορίας και αν κάποια έχει χαμηλότερο κόστος από το σχετικό του ιμπεριαλιστικό κράτος, τότε ανταλλάσσουν θέσεις.

10. *Έλεγχος συνολικού κόστους κάθε αυτοκρατορίας..*

    Το κόστος κάθε αυτοκρατορίας εξαρτάται από την δύναμη των ιμπεριαλιστικών κρατών, αλλά και των αποικιών τους.

11. *Εφαρμογή του ιμπεριαλιστικού ανταγωνισμού.*

    Όλες οι αυτοκρατορίες με βάση την δύναμή τους (το συνολικό κόστος) προσπαθούν να λάβουν υπό την κατοχή τους τις αποικίες της ασθενέστερης αυτοκρατορίας.

12. *Αφαίρεση της ασθενέστερης αυτοκρατορίας.*

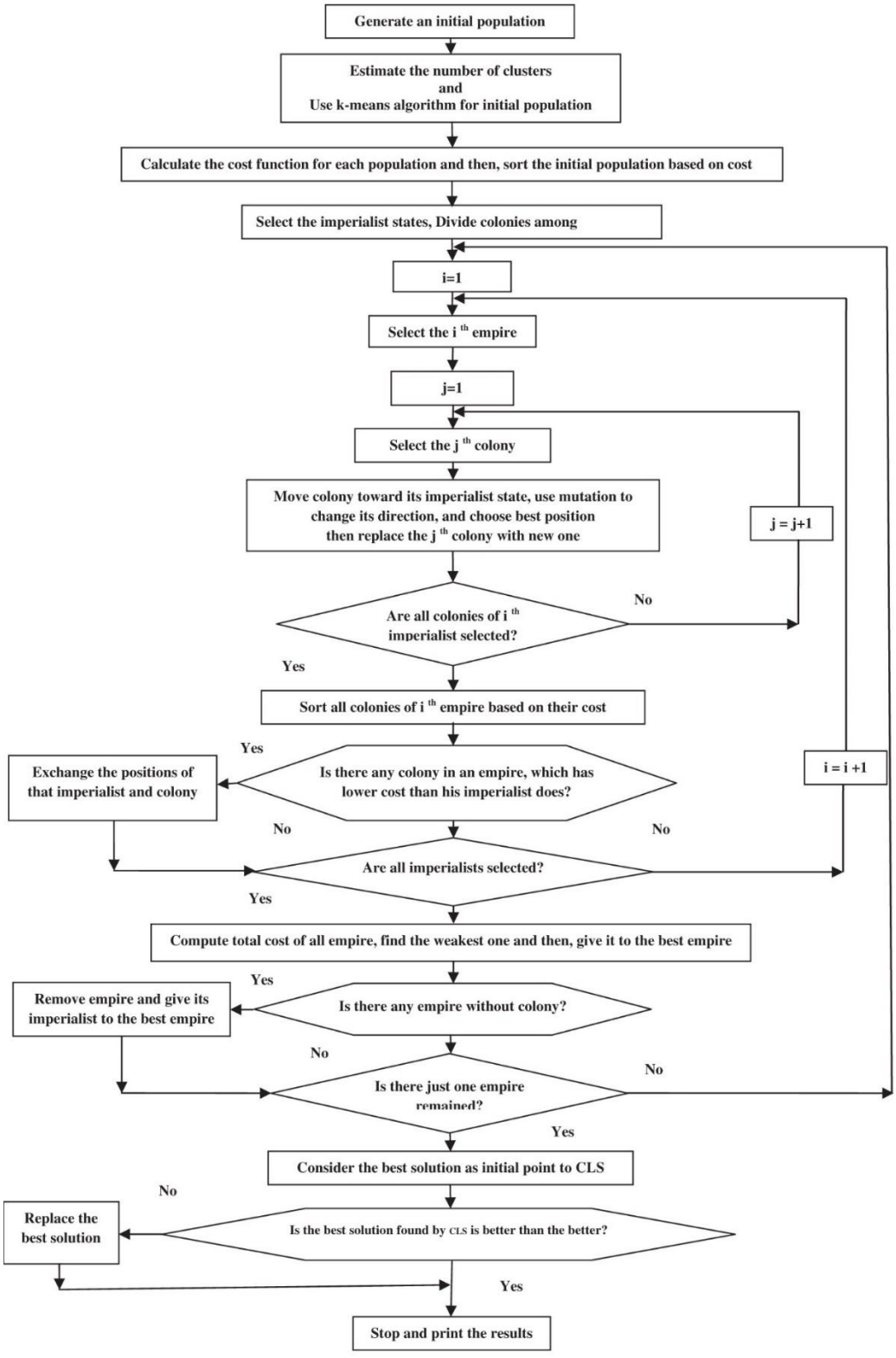
    Εάν υπάρχει κάποια αυτοκρατορία χωρίς αποικία, απαλείφεται. Μια από τις ασθενέστερες αποικίες της καλύτερης αυτοκρατορίας (με το χαμηλότερο κόστος) αντικαθίσταται με αυτόν τον ιμπεριαλιστή.

13. *Εφαρμογή της chaotic local search (CLS) για την αναζήτηση ολικής (global) λύσης.*

    Στον CLS αλγόριθμο θεωρείται ως αρχική λύση η καλύτερη λύση. Εάν η λύση που υπολογίσει ο CLS είναι καλύτερη από την ήδη υπάρχουσα, τότε εκείνη αντικαθίσταται.

14. *Έλεγχος του αριθμού των αυτοκρατοριών.*

    Εάν απομένει μόνο μία αυτοκρατορία, ο αλγόριθμος τερματίζει. Αλλιώς πηγαίνει στο βήμα 7.



6. Διάγραμμα ροής του Hybrid k-MICA αλγορίθμου

Στα πειράματα που διεξήχθησαν, ο Hybrid k-MICA συγκρίθηκε με διάφορους στοχαστικούς αλγορίθμους, όπως οι MICA, ICA, ACO, PSO, SA, GA, TS, HBMO και k-means. Από τα αποτελέσματα, ο προτεινόμενος Hybrid k-MICA μπορεί να θεωρηθεί βιώσιμη και αποδοτική μέθοδος για την εύρεση της βέλτιστης ή σχεδόν βέλτιστης λύσης σε προβλήματα ομαδοποίησης. Επίσης, κρίθηκε συγκρίσιμος με τους υπόλοιπους αλγορίθμους από την άποψη των καλύτερων, μέσων και χειρότερων λύσεων και της τυπικής απόκλισης. Πέρα από την ισχύ και την αποδοτικότητά του, ο Hybrid k-MICA μπορεί να εφαρμοστεί μόνο όταν ο αριθμός των clusters είναι γνωστός εκ των προτέρων.

4.2.10 Clustering Massive Small Data for IOT

Το Hadoop Distributed File System (HDFS) έχει σχεδιαστεί για την αποθήκευση και διαχείριση μεγάλου όγκου συνόλων δεδομένων. Η προεπιλεγμένη μονάδα αποθήκευσης είναι τα 64 MB. Στην πράξη, όμως, τα περισσότερα δεδομένα που παράγονται είναι μικρότερα από 64 MB, γεγονός που οδηγεί σε σπατάλη της μνήμης και μείωση της απόδοσης του συστήματος. Σκοπός είναι, λοιπόν, η βελτίωση της απόδοσης της επεξεργασίας δεδομένων, όπως και η βελτίωση του ποσοστού χρησιμοποίησης του συστήματος. Η κύρια ιδέα για την επίτευξη αυτού του σκοπού είναι να συγχωνευτούν τα «μικρά δεδομένα» (small data) και να σχηματίσουν μεγαλύτερα σύνολα δεδομένων. Σε αυτή τη δημοσίευση, **"*Clustering Massive Small Data for IOT*"** [39], εφαρμόζεται κυρίως η Sequence File στρατηγική ομαδοποίησης και παρουσιάζεται μια στρατηγική συγχώνευσης βασισμένη στο MapReduce. Πρώτα, χρησιμοποιείται ο k-means για την ομαδοποίηση των μεγάλων συνόλων «μικρών δεδομένων» που έχουν ταιριαστά χαρακτηριστικά και ύστερα, μια στρατηγική συγχώνευσης αρχείων. Τα μεγάλα, συγχωνευμένα, σύνολα δεδομένων, μπορούν να χειριστούν με ενιαίες λειτουργείες και βελτιώνουν την απόδοση ανάκτησης δεδομένων. Με κατάλληλα πειράματα στο Hadoop, αποδείχτηκε η σημαντική βελτίωση στην απόδοση του συστήματος.

4.2.11 Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion

Η ομαδοποίηση βρίσκει εφαρμογή και στην αναζήτηση αποτελεσμάτων στο διαδίκτυο. Τα συστήματα που εκτελούν αυτή τη διαδικασία ονομάζονται Μηχανές Ομαδοποίησης Διαδικτύου (Web Clustering Engines). Οι μηχανές αυτές λειτουργούν ως μετα-μηχανές, δηλαδή συλλέγουν έναν σημαντικό αριθμό αποτελεσμάτων από τις συμβατικές μηχανές αναζήτησης και στη συνέχεια τα ομαδοποιούν

δημιουργώντας κάποιες ειδικότερες λίστες. Με αυτόν τον τρόπο ο χρήστης μπορεί να φιλτράρει μεγάλο αριθμό αποτελεσμάτων και να επιταχύνει την αναζήτησή του.

Τα δύο επικρατέστερα προβλήματα που υπάρχουν στις μηχανές ομαδοποίησης διαδικτύου είναι οι ασυνέπειες (inconsistencies) στο περιεχόμενο των clusters και οι ασυνέπειες στην περιγραφή του cluster. Στο πρώτο πρόβλημα το περιεχόμενο των clusters δεν ανταποκρίνεται πάντα στην ετικέτα (label). Επίσης, η πλοήγηση μέσω των ιεραρχιών δεν οδηγεί απαραίτητα σε πιο ειδικά αποτελέσματα. Το δεύτερο πρόβλημα αναφέρεται στην ανάγκη για πιο εκφραστικές περιγραφές των clusters. Τα δύο αυτά προβλήματα αποτέλεσαν το κίνητρο για την παρούσα εργασία [40], στην οποία αναπτύσσεται ένας νέος αλγόριθμος βασισμένος στους Cuckoo search (CS) metaheuristic, k-means, Balanced Bayesian Information Criterion (BBIC), τις split και merge μεθόδους σε clusters και την συχνή διατύπωση (frequent phrases approach) για την ανάθεση ετικετών σε clusters. Πρόκειται για την πρώτη φορά που οι παραπάνω μέθοδοι χρησιμοποιήθηκαν συνδυαστικά.

Ο νέος αλγόριθμος ονομάζεται ***Web Document Clustering based on the Cuckoo Search Algorithm (WDC-CSK)*** [40] και είναι ένας description-centric αλγόριθμος βασισμένος στον Cuckoo Search (CS) [41]. Ο CS είναι εμπνευσμένος από τον αναπαραγωγικό παρασιτισμό (brood parasitism) κάποιων ειδών κούκων* σε συνδυασμό με την συμπεριφορά των πτήσεων Lévy ορισμένων πτηνών και μυγών [42]. *[Οι θηλυκοί κούκοι γεννούν τα αυγά τους στις φωλιές άλλων πτηνών και στη συνέχεια, όταν τα πτηνά-ξενιστές (host birds) ανακαλύψουν τα ξένα αυγά, είτε τα απορρίπτουν είτε εγκαταλείπουν τη φωλιά τους και χτίζουν καινούρια κάπου αλλού]. Ο k-means χρησιμοποιήθηκε ως τοπική στρατηγική για τη βελτίωση των ολικών (global) λύσεων του CS. Οι πτήσεις Lévy [43] αντικαθίστανται από τις μεθόδους split και merge, οι οποίες χρησιμοποιούνται για την προώθηση της ποικιλίας στον πληθυσμό και για να τον αποτρέψουν απ' το να συγκλίνει πολύ γρήγορα σε τοπικά βέλτιστες λύσεις. Τέλος, ως συνάρτηση καταλληλότητας (fitness function), η οποία βοηθά τον αλγόριθμο να βρει αυτόματα τον αριθμό των clusters, χρησιμοποιείται είτε το Balanced Bayesian Information Criterion (BBIC) είτε το Bayesian Information Criterion (BIC) – αν και προτείνεται το BBIC. Μια σύνοψη του WDC-CSK φαίνεται στην εικόνα 7.

```
01  Initialize algorithm parameters
02  Document preprocessing
03  Execute in parallel a specific number (MNI) of Islands
04      Initialize population of nests; create randomly a set of nests (population of nests) from the current island
05      Execute k-means (local optimizer) for each nest in population from the current island
06      Calculate fitness values (BBIC or BIC) according to (4) or (5) for all nests in population from the current island
07      Repeat
08          Create a new nest using abandon, split or merge operations (methods) based on a randomly selected nest
            (current nest) from the current island
09          Execute k-means (local optimizer) for the new generated nest
10          Calculate fitness value (BBIC or BIC) according to (4) or (5) for the new generated nest
11          Store best solution, if the new generated nest is better than another randomly selected nest, this last nest is
            replaced in the population for the new generated nest
12      Until stopping conditions are satisfied (MNN parameter is reached or MET parameter is reached)
13      Select the best nest in the population of nest from the current island
14  End on parallel execution
15  Select the best nest from all islands
16  Assign labels to clusters in the best nest based on the frequent phrases in each cluster.
```

7. WDC-CSK αλγόριθμος

```
01  Select an Initial Partition (k centers)
    Repeat
02      Data Assignment: Re-compute Membership
03      Relocation of "means": Update Centers
04  Until (Stop Criterion)
05  Return Solution
```

8. k-means αλγόριθμος

Τα σημαντικότερα βήματα του WDC-CSK είναι τα παρακάτω:

01.

Αρχικοποίησε τις παραμέτρους του αλγορίθμου. Οι παράμετροι που χρειάζεται ο WDC-CSK για την ελαχιστοποίηση του BBIC κριτηρίου είναι οι εξής:

- Ο Μέγιστος Αριθμός Νησιών (The Maximum Number of Islands – MNI) – ένας ακέραιος αριθμός μεταξύ 1 και 5
- Το Μέγεθος Πληθυσμού (Population Size – PS) – ένας ακέραιος αριθμός μεταξύ 5 και 10
- Η Αντικειμενική Συνάρτηση (Objective Function – OF) – μια τιμή απαρίθμησης μεταξύ BBIC και BIC
- Η τιμή της Πιθανότητας μιας Εγκαταλελειμμένης φωλιάς (Probability value of Abandoned nest – PA) – μια πραγματική τιμή μεταξύ 0.1 και 0.2

- Term Frequency Threshold – (TFT) – μια ακέραια τιμή μεγαλύτερη από 2 για την διαδικασία ανάθεσης ετικετών
- Ο Μέγιστος Αριθμός Κύκλων (Maximum Number of Cycles – MNCK) που απαιτείται από τον k-means για να συγκλίνει – ένας ακέραιος μεγαλύτερος ή ίσος με 1
- Ο Μέγιστος Αριθμός Φωλιών (Maximum Number of Nests – MNN) ή ο Μέγιστος Χρόνος Εκτέλεσης (Maximum Execution Time – MET) σε milliseconds, ως κριτήριο τερματισμού του αλγορίθμου

Η MNI παράμετρος και οι γραμμές 03 και 14 επιτρέπουν στον WDC-CSK να εκτελεί μια διαδικασία αναζήτησης παράλληλα χρησιμοποιώντας νήματα (threads) που δεν μοιράζονται πληροφορίες, γνωστά ως "islands". Η MNI παράμετρος καθορίζει τον αριθμό των νημάτων (islands) που ο WDC-CSK εκτελεί χωριστά και παράλληλα. Οι γραμμές 04 ως 13 εκτελούνται ως μονάδα σε κάθε island (νήμα εκτέλεσης). Όταν όλα τα νήματα έχουν τελειώσει το έργο τους, ο αλγόριθμος επιλέγει την καλύτερη φωλιά που βρέθηκε από όλα τα islands.

02.

Προεπεξεργασία εγγράφου. Το στάδιο αυτό περιλαμβάνει tokenization (λεξιλογική ανάλυση), φιλτράρισμα πεζών γραμμάτων, αποκοπή καταλήξεων με τον αλγόριθμο του Porter και την κατασκευή της Term-by-Document Matrix (TDM) η οποία χρησιμοποιείται συχνά για την αναπαράσταση εγγράφων κατά την ανάκτηση πληροφορίας.

04.

Αρχικοποίησε τον πληθυσμό της φωλιάς. Στον WDC-CSK οι φωλιές αναπαριστούν τις λύσεις. Κάθε φωλιά έχει διαφορετικό αριθμό clusters, μια λίστα κεντροειδών και την τιμή της αντικειμενικής συνάρτησης με βάση το BBIC ή το BIC που εξαρτάται από την τοποθεσία και τον αριθμό των κεντροειδών σε κάθε φωλιά. Αρχικά, κάθε κεντροειδές αντιστοιχεί σε ένα τυχαία επιλεγμένο έγγραφο στην TDM μήτρα. Ο αρχικός αριθμός των clusters, k, υπολογίζεται επίσης τυχαία.

05.

Εκτέλεσε τον k-means. Οι γραμμές 02 με 05 στο σχήμα 8 εκτελούνται με βάση τα καταγεγραμμένα κεντροειδή κάθε φωλιάς. Η εκτέλεσή τους επαναλαμβάνεται MNCK φορές. Αυτή η παράμετρος ελέγχει το επίπεδο αξιοποίησης του αλγορίθμου. Εάν είναι ίση με 1, ο αλγόριθμος δεν βελτιώνει τις λύσεις, παρά μόνο οργανώνει τα έγγραφα στα κατάλληλα κεντροειδή και τα υπολογίζει ξανά. Εάν η τιμή της είναι μεγαλύτερη από 1, ο αλγόριθμος

βελτιώνει τοπικά τις λύσεις. Παρόλα αυτά, υπάρχει περίπτωση ο αλγόριθμος να συγκλίνει σε λιγότερους κύκλους.

08.

Δημιούργησε μια νέα φωλιά. Για τη δημιουργία μιας νέα φωλιάς (λύσης) ο αλγόριθμος εκτελεί μια πράξη abandon, merge ή split. Με μια συγκεκριμένη πιθανότητα που ορίζεται από την παράμετρο PA, ο αλγόριθμος δημιουργεί μια νέα φωλιά με τυχαία επιλεγμένα κεντροειδή από την TDM μήτρα. Αυτή η πράξη αντιστοιχεί σε abandon και είναι εμπνευσμένη από την κατάσταση στην οποία ένα αυγό κούκου ανακαλύπτεται από το πτηνό-ξενιστή. Σε αυτή την περίπτωση, δημιουργείται μια εντελώς καινούρια φωλιά για να ολοκληρώσει τον πληθυσμό των φωλιών του κούκου στο τρέχον νησί. Αυτή η πράξη παρέχει ποικιλομορφία και αποτρέπει τον πληθυσμό φωλιών να συγκλίνει πολύ γρήγορα. Με μια συγκεκριμένη πιθανότητα $((1 - PA) * 0.5)$ εκτελείται η πράξη split ή merge. Αυτές οι πράξεις αντικαθιστούν τις πτήσεις Lévy του πρωτότυπου Cuckoo Search αλγορίθμου. Και για τις δύο πράξεις, αρχικά επιλέγεται τυχαία μια φωλιά από τον τρέχοντα πληθυσμό. Αυτή η φωλιά αντιγράφεται σε μια νέα φωλιά και ονομάζεται “base nest”. Στην πράξη merge, τα δύο περισσότερο παρόμοια κεντροειδή (μετρημένα με την ομοιότητα συνημιτόνου) από την “base nest” επιλέγονται και ενσωματώνονται. Στην πράξη split, ο πιο διασκορπισμένος cluster επιλέγεται (με βάση το SSE) και χωρίζεται σε δύο clusters. Ο νέος cluster σχηματίζεται επιλέγοντας για κεντροειδές το πιο διαφορετικό έγγραφο του πιο διασκορπισμένου cluster.

13.

Επίλεξε την καλύτερη φωλιά. Σε αυτό το βήμα ο αλγόριθμος βρίσκει και επιλέγει την καλύτερη λύση στον πληθυσμό των φωλιών από το τρέχον νησί. Η καλύτερη φωλιά είναι εκείνη με την χαμηλότερη τιμή καταλληλότητας (fitness). Αυτή η λύση επιστρέφεται ως η καλύτερη λύση για την ομαδοποίηση (κεντροειδή και καταλληλότητα) από το τρέχον νησί.

16.

Ανάθεσε ετικέτες στις ομάδες. Ο αλγόριθμος χρησιμοποιεί μια Frequent Phrases (FPH) μέθοδο για την ανάθεση ετικετών στους clusters. Αυτό το βήμα αντιστοιχεί με το βήμα “Frequent Phrase Extraction” στον αλγόριθμο Lingo [44] (με κάποιες τροποποιήσεις). Στον WDC-CSK αυτή η μέθοδος χρησιμοποιείται για κάθε cluster που έχει παραχθεί στις καλύτερες λύσεις. Η ανάθεση ετικετών λειτουργεί ως εξής:

i) **Μετατροπή της αναπαράστασης.** Όλα τα έγγραφα στον τρέχοντα cluster επιλέγονται και η αναπαράστασή τους μετατρέπεται από character-based σε word-based.

ii) **Σύνδεση εγγράφων.** Τα έγγραφα συνδέονται σε μια αλληλουχία και δημιουργείται ένα νέο έγγραφο με την ανεστραμμένη εκδοχή της αλληλουχίας.

iii) **Ανακάλυψη πλήρων φράσεων.** Στον τρέχοντα cluster, ανακαλύπτονται οι πλήρεις φράσεις αριστερά και δεξιά, ταξινομούνται αλφαβητικά και συνδυάζονται σε ένα σύνολο ολοκληρωμένων φράσεων.

iv) **Τελική επιλογή.** Οι όροι και οι φράσεις που βρίσκονται στον τρέχοντα cluster και ξεπερνούν το Term Frequency Threshold (TFT), επιλέγονται. Οι όροι αναζήτησης του χρήστη αφαιρούνται από τους επιλεγμένους όρους και φράσεις ώστε να βελτιωθεί η ποιότητα της διαδικασίας ανάθεσης ετικετών.

v) **Κατασκευή της ετικέτας και της ομάδας «Άλλα».** Ο αλγόριθμος χρησιμοποιεί το TFT στα έγγραφα και αν κάποια δε το φτάνουν, στέλνονται σε άλλους clusters.

vi) **Συμπέρασμα για την ετικέτα ομάδας.** Για τους τρέχοντες clusters κατασκευάζεται μια term-by-document μήτρα και ύστερα χρησιμοποιείται η ομοιότητα συνημιτόνου για την εύρεση των περισσότερο παρόμοιων υποψήφιων όρων ή φράσεων για τον cluster.

Η ολική πολυπλοκότητα του WDC-CSK αλγορίθμου είναι $O(MNI * (PS + MNN) * n * k * MNCK)$ η οποία είναι γραμμική σε σχέση με το n, τον συνολικό αριθμό εγγράφων στη συλλογή.

Ο WDC-CSK συγκρίθηκε με τον Suffix Tree Clustering (STC) [45] και τον Lingo [44] από δύο απόψεις: 1) την ποιότητα των ομαδοποιημένων αποτελεσμάτων και 2) την ευκολία με την οποία οι χρήστες μπορούν να χρησιμοποιήσουν τα ομαδοποιημένα αποτελέσματα. Ο STC είναι ο πρώτος αλγόριθμος ομαδοποίησης για αναζήτηση στο διαδίκτυο που βασίστηκε στα δέντρα καταλήξεων (suffix trees) και τις συχνές φράσεις (frequent phrases). Ο Lingo είναι γνωστός διάδοχος του STC, στον οποίο οι συχνές φράσεις των εγγράφων εξάγονται πρώτα με χρήση πινάκων καταλήξεων (suffix arrays), ύστερα με Singular Value Decomposition (SVD) επιλέγονται οι καλύτερες συχνές φράσεις και τέλος τα έγγραφα κατανέμονται σε αυτές τις συχνές φράσεις. Ο WDC-CSK συγκρίθηκε επίσης με τους Lingo3G, KeySRC, OPTIMSRC και το Yahoo!. Ο Lingo3G χρησιμοποιεί ένα custom-built meta-heuristic για την επιλογή καλά ορισμένων και ποικίλων ετικετών για τους clusters. Ο KeySRC είναι μια μηχανή που έχει χτιστεί πάνω στον STC, ο OPTIMSRC ένας αλγόριθμος ομαδοποίησης web εγγράφων ο οποίος δημιουργεί τα meta-partitions με “stochastic hill climbing” ακολουθούμενο από meta-labeling με βάση ετικέτες των Lingo,

STC και KeySRC. Τα αποτελέσματα του Yahoo! είναι τα πρωτότυπα αποτελέσματα που επέστρεψε η γνωστή μηχανή αναζήτησης.

Ο WDC-CSK επιτυγχάνει καλύτερο αριθμό clusters σε όλα τα σύνολα δεδομένων του πειράματος και με σημαντική διαφορά. Κατά μέσο όρο, ο WDC-CSK διαφέρει από τον ιδανικό αριθμό clusters κατά 0.93-1.11 ομάδες, ενώ ο Lingo κατά 20.28, ο STC κατά 6.73 και ο Bisecting k-means κατά 3.74. Επίσης, ο WDC-CSK παρουσιάζει καλύτερα αποτελέσματα από τους άλλους τρεις αλγορίθμους (Lingo, STC, Bisecting k-means) στα εξής μέτρα αξιολόγησης: Precision, Recall, F-measure, Fall-out και Accuracy (Rand index). Με χρήση του Subtopic Search Length under k document sufficiency (SSL_k) αξιολογήθηκε η ευκολία με την οποία οι χρήστες έκαναν χρήση των αποτελεσμάτων της ομαδοποίησης και ο WDC-CSK είναι μια βελτίωση σε όλους τους αλγορίθμους. Ο WDC-CSK παρουσιάζει εξαιρετικά αποτελέσματα στα σύνολα δεδομένων αναφοράς και σε σύγκριση με τους διάφορους state of the art αλγορίθμους δείχνει βελτίωση μεταξύ 5.86% και 35.89% στο F-measure, μεταξύ 6.02% και 43.79% στο Recall, μεταξύ 3.67% και 6.82% στο Accuracy και μεταξύ 2.52% και 45.39% στο Fall-out. Τέλος, τα πειράματα έδειξαν βελτίωση και των τιμών SSL_k μεταξύ 21.70% και 31.76%.

4.2.12 GGSA: A Grouping Gravitational Search Algorithm for data clustering

Ο Gravitational Search Algorithm (GSA) [46] είναι ένας στοχαστικός, μετα-ευρετικός (metaheuristic), βασισμένος στον πληθυσμό (population-based) αλγόριθμος, σχεδιασμένος για την επίλυση προβλημάτων συνεχούς βελτιστοποίησης. Πρόκειται για μια τεχνική βελτιστοποίησης σμήνους (swarm optimization technique) η οποία προσομοιώνει τις αλληλεπιδράσεις μεταξύ αντικειμένων (ερευνητών agents), οι οποίοι είναι ένα σύνολο μαζών, με βάση τους νόμους του Νεύτωνα για την βαρύτητα και την κίνηση. Ένα σύνολο αντικειμένων (agents), εισάγεται στον D-διαστάσεων χώρο των λύσεων (solution space) του προβλήματος για να βρει τη βέλτιστη λύση. Η θέση κάθε agent στον GSA επιδεικνύει μια υποψήφια λύση του προβλήματος. Γι' αυτό κάθε agent αναπαρίσταται από ένα διάνυσμα στον χώρο αυτό. Η αποδοτικότητα των agents μετριέται με την τιμή της «μάζας» τους (mass value), δηλαδή οι μάζες με μεγάλο βάρος αντιστοιχούν σε καλές λύσεις. Η μέθοδος δρα επαναληπτικά, και με την πάροδο του χρόνου, οι μάζες έλκονται από την βαρύτερη μάζα. Με αυτόν τον τρόπο υποδεικνύεται η βέλτιστη λύση. Ο ψευδοκώδικας του αλγορίθμου φαίνεται στην εικόνα 9.

```
Generate the initial population;
Evaluate the fitness value for each agent;
Calculate the mass value for each agent;
While stopping criteria is not satisfied Do
    Update G, K, and Kbest;
    Calculate the acceleration of each agent by Eq. (9);
    Calculate the velocity of each agent by Eq. (11);
    Update the position of each agent by Eq. (12);
    Evaluate the fitness for each agent;
    Calculate the mass value for each agent;
Endwhile
Output: Best solution found.
```

9. Ψευδοκώδικας του απλού GSA αλγορίθμου

Ο ***Grouping Gravitational Search Algorithm (GGSA)*** [47] είναι μια προσαρμογή της δομής του απλού GSA με σκοπό την επίλυση προβλημάτων ομαδοποίησης. Γι' αυτό το λόγο, προτείνεται ένα ειδικό encoding scheme ώστε να ληφθεί υπόψη η δομή αυτών των προβλημάτων. Δεδομένου του encoding αναπτύχθηκαν και νέες εξισώσεις για τον υπολογισμό της επιτάχυνσης, της ταχύτητας και της θέσης του κάθε agent, διατηρώντας μεν τα κύρια χαρακτηριστικά των ήδη υπαρχόντων, καθιστώντας δε εφικτή τη λειτουργεία τους με clusters δεδομένων. Το κύριο χαρακτηριστικό των νέων εξισώσεων είναι πως λειτουργούν σε συνεχή χρόνο, αλλά το αποτέλεσμα χρησιμοποιείται στον χώρο τον clusters (cluster space) μέσω μιας διαδικασίας δύο φάσεων. Πιο συγκεκριμένα, οι τελεστές (operators) που πρέπει να επαναπροσδιοριστούν είναι: ο τελεστής γραμμικής απόστασης ("–"), ο τελεστής Ευκλείδειας απόστασης και ο τελεστής κίνησης ("+"). Η γραμμική απόσταση αντικαθίσταται από την απόσταση μεταξύ δύο clusters, για τον υπολογισμό της οποίας επιλέχθηκε η απόσταση Jaccard. Για τον επαναπροσδιορισμό της Ευκλείδειας απόστασης πρέπει να ληφθεί υπόψη ότι οι clusters στον GGSA παίζουν το ρόλο των μεταβλητών στον GSA. Δηλαδή, ενώ στον GSA η θέση του αντικειμένου στην d-οστή διάσταση αντιπροσωπεύει την τιμή της d-οστής μεταβλητής, στον GGSA καθορίζει τα δεδομένα που ανήκουν στον d-οστό cluster.

Η απόδοση του GGSA συγκρίθηκε με την απόδοση των εξής αλγορίθμων: Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Firefly Algorithm (FA) και εννέα ακόμα γνωστές τεχνικές ομαδοποίησης (Bayes Net, Multi Layer Perceptron Artificial Neural Network (MPL-ANN), Radial Basis Function Artificial Neural Network (RBF-ANN), KStar, Bagging, MultiBoostAB, Naïve Bayes Tree (NBTree), Ripple Down Rule (Ridor) και Voting Feature Interval (VFI)). Τα αποτελέσματα

επιβεβαιώνουν την αποτελεσματικότητα του GGSA και δείχνουν ότι μπορεί να εφαρμοστεί επιτυχώς για την ομαδοποίηση δεδομένων. Η σύγκριση με τους παραπάνω αλγορίθμους έγινε με βάση το μέσο τους Classification Error Percentage (CEP) που προέκυψε από πειράματα σε 13 σύνολα δεδομένων-σημεία αναφοράς, όπου CEP:

$$CEP = 100 \times \frac{Αριθμός\ εσφαλμένα\ ταξινομημένων\ περιστατικών\ (instances)}{Συνολικό\ μέγεθος\ του\ συνόλου\ δοκιμής}$$

Από τα ποσοστά του μέσου CEP, προέκυψε μια κατάταξη στην οποία ο GGSA κατατάσσεται στην 1η θέση – δηλαδή κατέχει το μικρότερο ποσοστό εσφαλμένης ταξινόμησης (CEP). Μάλιστα, η απόδοσή του σε σχέση με τον PSO και τον GSA ήταν καλύτερη και στα 13 σύνολα δεδομένων, ενώ με τον ABC στα 10 από αυτά.

4.3 Στόχος: Μείωση κατανάλωσης ενέργειας

4.3.1 An Energy Balanced Cluster Algorithm for Wireless Sensor Networks

Τα Ασύρματα Δίκτυα Αισθητήρων (WSN – Wireless Sensor Networks) είναι multi-hop δίκτυα που αποτελούνται από πολλούς κόμβους-αισθητήρες κατανεμημένους σε μια γεωγραφική περιοχή με σκοπό την παρατήρηση και καταγραφή φυσικών μεγεθών ή περιβαλλοντικών συνθηκών (όπως θερμοκρασία, ήχο, πίεση κ.ά.). Η πηγή ενέργειας των αισθητήρων είναι συχνά μια μπαταρία με περιορισμένο προϋπολογισμό ενέργειας. Επίσης, λόγω του ότι τα WSNs αναπτύσσονται σε μη βολικά ή δύσκολα προσβάσιμα σημεία, η επαναφόρτιση των μπαταριών είναι συχνά δύσκολη ή αδύνατη. Η διάρκεια ζωής του δικτύου πρέπει να είναι αρκετή προκειμένου να πληρούνται οι απαιτήσεις της εφαρμογής. Έτσι, η έρευνα έχει επικεντρωθεί στο σχεδιασμό ενός ενεργειακά ισορροπημένου αλγορίθμου ομαδοποίησης για δρομολόγηση (routing). Ένας τέτοιος αλγόριθμος αποτελείται από τρία στάδια: την κατασκευή των clusters (cluster building), την επιλογή διαδρομής μετάδοσης (selection of transmission path) και την επικοινωνία του δικτύου (network communication). Σε έναν καλό αλγόριθμο ομαδοποίησης θα πρέπει ο χρόνος της επικοινωνίας του δικτύου να είναι πολύ μεγαλύτερος από τον χρόνο κατασκευής των clusters.

Ο *Low Energy Adaptive Clustering Hierarchy (LEACH)* [48] ήταν ο πρώτος αλγόριθμος που χρησιμοποιήθηκε στα WSNs και επιλέγει τυχαία cluster-heads (οι οποίοι λειτουργούν ως συντονιστές) για να διανεμηθεί ομοιόμορφα ο ενεργειακός φόρτος (energy load) μεταξύ των αισθητήρων του δικτύου. Ο *Deterministic Cluster-Head Selection (DCHS)* [49] είναι ένας βελτιωμένος αλγόριθμος βασισμένος στον

LEACH που χρησιμοποιεί την παράμετρο της ενέργειας για να επηρεάσει την επιλογή cluster-head έτσι ώστε να εγγυηθεί ότι οι κόμβοι έχουν αρκετή ενέργεια για να λειτουργήσουν ως cluster-heads. Επίσης, εξισορροπεί ως ένα βαθμό την κατανάλωση ενέργειας. Έχουν αναπτυχθεί και άλλοι αλγόριθμοι δρομολόγησης με ομάδες (clustering routing), υπάρχουν όμως ελλείψεις. Έτσι, οι ερευνητές στο ***"An Energy Balanced Cluster Algorithm for Wireless Sensor Networks"*** [50] ανέπτυξαν έναν αλγόριθμο με βάση τα σημεία που θεωρούσαν ότι δεν είχε προηγουμένως δοθεί βάση:

- Το πρόβλημα "bottleneck" στους cluster-heads
- Την αποτελεσματική και ισορροπημένη χρήση της ενέργειας
- Την παράταση της διάρκειας ζωής των WSNs.

Το μονοπάτι inter-cluster επικοινωνίας και η μέθοδος επιλογής cluster-heads εγγυώνται ομοιόμορφη κατανομή των cluster-heads και μείωση της πιθανότητας να εμφανιστεί το φαινόμενο "bottleneck", ενώ οι προτεινόμενες μέθοδοι για την ανακατασκευή των clusters (cluster rebuilding) και την ενδο-επικοινωνία στους clusters (intra-cluster communication) στοχεύουν στην αύξηση της αξιοποίησης της διαθέσιμης ενέργειας και την παράταση της διάρκειας ζωής των δικτύων.

Επιλογή Cluster-Head

Σκοπός είναι η επίτευξη καλής ενεργειακής απόδοσης από την άποψη διάρκειας ζωής του δικτύου και ομοιόμορφης κατανομής αυτής, όχι απλά από άποψη ενεργειακής κατανάλωσης. Κατά την επιλογή του cluster-head, κάθε κόμβος ορίζει έναν τυχαίο αριθμό μεταξύ 0 και 1. Αν ο αριθμός είναι μικρότερος από το κατώφλι $T(i)$, τότε ο κόμβος γίνεται υποψήφιος cluster-head και υπολογίζει τον αριθμό των γειτονικών κόμβων $Nei(i).Num$ και τον μέσο αριθμό γειτονικών κόμβων $Nei.average$. Όταν ικανοποιείται η ανισότητα $Nei(i).Num > Nei.average$, ο κόμβος γίνεται cluster-head και προχωρά στην κατασκευή των clusters. Το κατώφλι ορίζεται ως:

$$T(i) = \frac{p}{1 - p[r \ mod(1/p)]} * \left[\frac{S(i).E}{E_{n\_max}} + \left(r_s \ div \frac{1}{p}\right)\left(1 - \frac{S(i).E}{E_{n\_max}}\right)\right]$$

όπου p είναι το επιθυμητό ποσοστό cluster-head, r είναι ο τρέχων γύρος/κύκλος (current round), $S(i).E$ η τρέχουσα ενέργεια, E_{n_max} η αρχική ενέργεια του κόμβου και r_s ο αριθμός διαδοχικών γύρων στους οποίους ένας κόμβος δεν έχει γίνει cluster-head. Ως γειτονικοί κόμβοι ορίζονται εκείνοι που απέχουν ένα βήμα (one-hop) και ο αριθμός γειτονικών κόμβων ορίζεται ως:

$$Nei.average = \frac{1}{N} * \sum_{i=1}^{N} Nei(i).Num$$

όπου N ο συνολικός αριθμός κόμβων του δικτύου και $Nei(i).Num$ ο αριθμός γειτονικών κόμβων.

Διαμόρφωση Cluster (Cluster Set-up)

Κατά την κατασκευή των clusters, κάθε κόμβος επιλέγει τον cluster-head με τη μεγαλύτερη τιμή "value". Αν το μέγιστο "value" εμφανίζεται πάνω από μία φορά, τότε ο κόμβος επιλέγει τον cluster-head με τους λιγότερους κόμβους-μέλη, ή εκείνον με την μέγιστη ενέργεια. Η "value" του cluster-head ορίζεται ως:

$$Value_{CH} = \frac{S(i).E + S_{CH}.E}{D_{N\_CH} + D_{CH\_BS}}$$

όπου $S_{CH}.E$ είναι η εναπομείνασα ενέργεια του cluster-head, D_{N_CH} η απόσταση ως τον cluster-head και D_{CH_BS} η απόσταση μεταξύ cluster-head και σταθμού βάσης (base station). Όταν ο cluster-head λάβει όλα τα μηνύματα από τους κόμβους-μέλη, δημιουργεί ένα TDMA (Time Division Multiple Access) χρονοδιάγραμμα (schedule) που υποδεικνύει σε κάθε cluster τη στιγμή που μπορεί να μεταδίδει δεδομένα. Αυτό το χρονοδιάγραμμα αναμεταδίδεται πίσω στους κόμβους του cluster. Μόλις δημιουργηθούν οι clusters και το TDMA χρονοδιάγραμμα έχει σταθεροποιηθεί, η μετάδοση δεδομένων μπορεί να ξεκινήσει.

Συνθήκες για Cluster Rebuilding

Προκειμένου να μειωθεί η κατανάλωση ενέργειας λόγω ανακατασκευής των clusters, προτού ξεκινήσει ο επόμενος γύρος επιλογής cluster-head, ο σταθμός βάσης υπολογίζει τη μέση ενέργεια των clusters και την εναπομείνασα ενέργεια του cluster-head. Αν ισχύει $E_{average} \geq 0.5 * E_{n\_max}$ και $S_{CH}.E \geq E_{th}$, δηλαδή αν ο ισχύων cluster-head μπορεί να διατηρήσει τη λειτουργία του δικτύου, τότε η μετάδοση δεδομένων ξεκινά. Αν όχι, οι clusters ανακατασκευάζονται. Η μέση ενέργεια ορίζεται ως $E_{average} = \frac{1}{N} * \sum_{i=1}^{N} S(i).E$ και το κατώφλι ενέργειας του *r*-οστού γύρου ως $E_{th} = \left[1 - \left(\frac{r}{n}\right)^2\right] * E_{n\_max}$, όπου n είναι ο συνολικός αριθμός γύρων της λειτουργίας του δικτύου.

Ο Αλγόριθμος Δρομολόγησης

Ο αλγόριθμος δρομολόγησης αποτελείται από την ενδο-επικοινωνία στους clusters (intra-cluster communication), δηλαδή την επικοινωνία μεταξύ κόμβων και cluster-heads, και την inter-cluster επικοινωνία μεταξύ των cluster-heads και του σταθμού βάσης.

Inter-cluster Επικοινωνία

Για την inter-cluster επικοινωνία, οι cluster-heads σχηματίζουν, στην ουσία, μια «αλυσίδα» επικοινωνίας ανάμεσα σε αυτούς και στον σταθμό βάσης. Κάθε cluster-head επιλέγει τον γειτονικό cluster-head με τη

μεγαλύτερη τιμή “value” ως κόμβο διαμετακόμισης και ύστερα από αυτό διαγράφεται από τη λίστα γειτονικών cluster-heads. Αν δεν μπορεί να βρει άλλον «γείτονα», τότε επικοινωνεί με τη βάση.

Intra-cluster Επικοινωνία

Προκειμένου να αποφευχθεί το πρόβλημα του “bottleneck”, (Το “bottleneck” έχει τη μικρότερη διακίνηση/throughput από όλα τα τμήματα του transaction path) δηλαδή η υπερφόρτωση ενός cluster-head, όταν η απόσταση μεταξύ κόμβου και cluster-head είναι μικρότερη από την κάλυψη της ακτίνας κόμβου $R_{compete}$, ο κόμβος επικοινωνεί με τον cluster-head. Αν όχι, η επικοινωνία ορίζεται ως εξής:

1. Εάν η σχέση μεταξύ εναπομείνασας και αρχικής ενέργειας του κόμβου ικανοποιεί την ανισότητα $S(i).E \geq 0.4 * E_{n\_max}$, πήγαινε στο βήμα 4. Αν όχι, πήγαινε στο βήμα 2.
2. Ο κόμβος επιλέγει έναν γειτονικό κόμβο που έχει τη μέγιστη απόσταση “Ratio” και ακολουθεί το βήμα 4. Αν υπάρχουν περισσότεροι από έναν κόμβοι με τη μέγιστη απόσταση “Ratio”, πήγαινε στο βήμα 3.
3. Ο κόμβος επιλέγει έναν γειτονικό κόμβο με τη μέγιστη τιμή $Value_{Nei}$ ως επόμενο βήμα (next hop) και μετά πήγαινε στο βήμα 4. Αν όχι, πήγαινε στο βήμα 2.
4. Οι κόμβοι επικοινωνούν με τον cluster-head και ο αλγόριθμος σταματά.

Σε αυτό το στάδιο, η ακτίνα κόμβου ορίζεται ως:

$$R_{compete} = \sqrt{\frac{M^2}{\pi * k_{opt}}}$$

όπου το k_{opt} ορίζεται ως:

$$k_{opt} = \sqrt{\frac{\varepsilon_{fs}}{\pi * (\varepsilon_{mp} + d^4 - E_{elec})}} * \frac{M * N}{2}$$

όπου E_{elec} είναι η ενέργεια που σπαταλιέται για τη λειτουργία του κυκλώματος πομπού ή δέκτη, ε_{mp} και ε_{fs} η ενέργεια που σπαταλιέται για τη λειτουργία του ενισχυτή μετάδοσης (transmission amplifier), M είναι το εύρος του δικτύου και το d^4 ορίζεται ως:

$$d^4 = \int_{y=0}^{y=M} \int_{x=x_{BS}-M}^{x=x_{BS}} \frac{\left(\sqrt{x^2 - (y - y_{BS})^2}\right)^4}{M^4} dxdy$$

Επίσης,

$$Ratio = \frac{S(i).E}{D_{N\_CH}}$$

και

$$Value_{Nei} = \frac{S(i).E + S_{Nei}.E}{D_{N\_Nei} + D_{Nei\_CH}}$$

όπου με $S_{Nei}.E$ συμβολίζεται η εναπομείνασα ενέργεια του γειτονικού κόμβου, με D_{N_Nei} η απόσταση μεταξύ κόμβου και γειτονικού κόμβου και με D_{Nei_CH} η απόσταση μεταξύ γειτονικού κόμβου και cluster-head.

Στο πείραμα προσομοιώθηκε ένα WSN αποτελούμενο από 100 κόμβους, τυχαία διανεμημένους σε ένα πεδίο $100m * 100m$ και ο βελτιωμένος αλγόριθμος συγκρίθηκε με τον DCHS [49] και τον "Opt Algorithm" [51] στον οποίο οι cluster-heads εναλλάσσονται εκ περιτροπής ώστε να κατανεμηθεί ομοιόμορφα τον ενεργειακό φόρτο (energy load) σε όλους τους κόμβους. Πρώτα συγκρίθηκαν ως προς την κατανάλωση ενέργειας. Με τον DCHS και τον Opt η ενέργεια του δικτύου εξαντλήθηκε ύστερα από 586 και 881 γύρους αντίστοιχα, ενώ με τον βελτιωμένο αλγόριθμο ύστερα από 1089, οπότε ο τελευταίος χρησιμοποιεί πιο αποτελεσματικά την ενέργεια του δικτύου. Ύστερα συγκρίθηκαν ως προς την αναλογία ανακατασκευής των clusters/διάρκεια ζωής του δικτύου. Ο Opt και ο DCHS είχαν ίση διάρκεια ζωής με τις φορές που έγινε ανακατασκευή των clusters, ενώ ο προτεινόμενος αλγόριθμος μειώνει σημαντικά τις φορές που πραγματοποιείται ανακατασκευή των clusters. Τέλος, παρατηρήθηκε ότι ο προτεινόμενος αλγόριθμος καθυστερεί σημαντικά τον «θάνατο» των κόμβων εξισορροπώντας την κατανάλωση ενέργειας και παρατείνει την διάρκεια ζωής του δικτύου.

4.3.2 An energy efficient hierarchical clustering index tree for facilitating time-correlated region queries in the Internet of Things

Στο Διαδίκτυο των Πραγμάτων, «έξυπνα» αντικείμενα (smart things) καταγράφουν φαινόμενα και επικοινωνούν μεταξύ τους. Τα ανιχνευμένα (sensed) δεδομένα συναθροίζονται/ομαδοποιούνται και οι συσκευές μετατρέπονται σε υπηρεσίες που ικανοποιούν «ερωτήματα» (queries) που θέτουν οι τελικοί χρήστες. Αυτή η συνάθροιση (aggregation) έχει αναγνωριστεί ευρέως ως αποτελεσματική μέθοδος μείωσης της κατανάλωσης ενέργειας σε ασύρματα δίκτυα αισθητήρων (WSNs). Με σκοπό, λοιπόν, την αποδοτική συλλογή και ομαδοποίηση δεδομένων, αναπτύχθηκε ένα ***Energy-Efficient Hierarchical Clustering index tree (ECH-tree)*** [52] βασισμένο στο "grid cell clustering" [53][54].

Στην τεχνική αυτή, πρώτα χωρίζεται ομοιόμορφα όλη η περιοχή του WSN σε κελιά, ώστε να δημιουργηθεί ένα πλέγμα (grid cells). Στη συνέχεια, τα κελιά αυτά ομαδοποιούνται εξασφαλίζοντας ότι η ενέργεια για την προώθηση μηνυμάτων μεταξύ τους θα είναι η ελάχιστη και αποτελούν μια «υπο-περιοχή». Αυτή η διαδικασία επαναλαμβάνεται μέχρι να σχηματιστεί ένα ιεραρχικό δέντρο, το *ECH-tree*. Η κατασκευή του ECH-δέντρου περιγράφεται στον Αλγόριθμο 1, στην οποία χρησιμοποιείται και η μέθοδος LEACH [55] για την επιλογή head nodes κάθε υπο-περιοχής, και στον Αλγόριθμο 2 η διαδικασία ομαδοποίησης των κελιών ή των υπο-περιοχών. Εκμεταλλευόμενοι την ιεραρχία του ECH-δέντρου, στην παρούσα δημοσίευση προτείνουν μια μέθοδο χρονικά συσχετισμένων ερωτημάτων περιοχής (time-correlated region queries) για την απάντηση συνεχών ερωτημάτων (όπως π.χ. η παρακολούθηση της υγρασίας μέσα σε ένα εργαστήριο), μετριάζοντας την κατανάλωση ενέργειας. Για τον υπολογισμό της ενέργειας που καταναλώνεται σε κάθε περίπτωση χρησιμοποιείται το ενεργειακό μοντέλο [56]. Τα βασικά βήματα αυτής της στρατηγικής είναι:

1. *Αρχική αναφορά δεδομένων αισθητήρα στον σταθμό βάσης:*
   Όταν ξεκινά να παρακολουθείται μια περιοχή, πρέπει οι τιμές όλων των κόμβων-αισθητήρων να συγκεντρωθούν και να αναφερθούν στο σταθμό βάσης μέσω του ECH-δέντρου. Ο σταθμός βάσης διατηρεί έναν πίνακα με όλες τις τρέχουσες τιμές των αισθητήρων στα κελιά.
2. *Συνεχής αναφορά δεδομένων αισθητήρα:*
   Στη συνέχεια, η τιμή κάποιου συγκεκριμένου αισθητήρα, θα διαβιβαστεί στον σταθμό βάσης μόνο εάν παρουσιάζει σημαντική διαφορά με την προηγούμενη τιμή που αναφέρθηκε, προκειμένου να εξοικονομηθεί ενέργεια.
3. *Συλλογή αποτελεσμάτων του ερωτήματος:*
   Τα ερωτήματα υποβάλλονται συνεχώς στον σταθμό βάσης, ο οποίος προσδιορίζει τα κελιά που έχουν σχέση με το ερώτημα αξιοποιώντας την ιεραρχία του ECH-tree. Τα ερωτήματα απαντώνται συναθροίζοντας τις τιμές των αισθητήρων στα κελιά ενδιαφέροντος. Ο Αλγόριθμος 3 περιγράφει τη διαδικασία απάντησης ενός ορισμένου ερωτήματος περιοχής (region query) από τον σταθμό βάσης και ο Αλγόριθμος 4 την συλλογή των δεδομένων από τα κελιά και τη διάθεση των αποτελεσμάτων στους χρήστες.

Στα πειράματα που πραγματοποιήθηκαν, αξιολογήθηκε η νέα μέθοδος σε σχέση με την παραδοσιακή, ως προς την καταναλισκόμενη ενέργεια και τον χρόνο ερωτήματος (query time) χρονικά συσχετισμένων ερωτημάτων περιοχής, για διάφορους αριθμούς αισθητήρων σε ίδιου μεγέθους περιοχή. Στις περισσότερες περιπτώσεις το ECH-tree κατανάλωνε λιγότερη ενέργεια από την παραδοσιακή μέθοδο και

ο χρόνος ερωτήματος ήταν επίσης λιγότερος. Σε λίγες περιπτώσεις παρατηρήθηκε ότι το ECH-tree κατανάλωνε περισσότερη ενέργεια, αλλά διαπιστώθηκε ότι αυτό συνέβαινε όταν η περιοχή ερωτήματος (query region) ήταν αραιή. Επομένως, τα αποτελέσματα έδειξαν ότι η νέα μέθοδος έχει καλύτερη απόδοση όταν οι αισθητήρες είναι περισσότεροι και πιο πυκνά τοποθετημένοι στη δοσμένη query region. Επίσης, με την αύξηση των κόμβων-αισθητήρων αυξάνεται και η καταναλισκόμενη ενέργεια και στις δύο μεθόδους. Αντίθετα, ο χρόνος ερωτήματος, είναι ανεξάρτητος του αριθμού των κόμβων αφού όταν οι επικεφαλής των κελιών του πλέγματος (grid cell headers) λάβουν το ερώτημα, το στέλνουν ταυτόχρονα στους κόμβους-αισθητήρες που περιέρχονται σε αυτά και όχι σε έναν-έναν. Τέλος, ορίζοντας ως διάρκεια ζωής του δικτύου το χρόνο έως τη στιγμή που ο πρώτος κόμβος θα εξαντλήσει την ενέργειά του, διαπιστώθηκε πως όσο πιο πυκνές ήταν οι query regions τόσο μεγαλύτερη ήταν η διάρκεια ζωής με την ECH-tree μέθοδο.

4.3.3 Design of an Improved Energy Efficient Clustering in M2M Communication

Η machine to machine (M2M) επικοινωνία είναι μια ανερχόμενη τεχνολογία που συνδέει πολλές M2M συσκευές, ενσωματωμένες με δυνατότητες δικτύωσης, ώστε να απαιτείται η ελάχιστη παρέμβαση από ανθρώπους. Στο [57] αναλύεται η machine to machine (M2M) επικοινωνία σε “Long Term Evolution for Machine-type communication” (LTE-M) δίκτυα, τα οποία παρέχουν τη διεπαφή (interface) στις M2M συσκευές ώστε να στέλνουν δεδομένα σε M2M πύλες (gateways) ή σε evolved Node B (eNB). Στα πλαίσια του Διαδικτύου των Πραγμάτων, όπου ο πληθυσμός M2M συσκευών που ελέγχονται από ένα eNB είναι 100.000 ή περισσότερες, είναι λογικό να υπάρξει συμφόρηση στο eNB και να μην μπορούν να έχουν πρόσβαση σε αυτό όλες οι συσκευές ταυτόχρονα. Γι’ αυτό απαιτείται ομαδοποίηση των συσκευών, η οποία αυξάνει και τη διάρκεια ζωής του δικτύου.

Ο ***Improved M2M Clustering Process (IMPC)*** αλγόριθμος [57] χρησιμοποιεί μια τροποποιημένη εκδοχή της εξίσωσης ενεργειακής απόδοσης (energy efficiency equation – EE) που παρουσιάζεται στην δημοσίευση [58], και είναι η εξής:

$$EE(p) = \frac{w_1 \log(1 + pA)}{\lambda \pi {R_c}^2 (C + pD)} + \frac{w_2 (p - p^2) \log\left(1 + \frac{B}{p}\right)}{C + pD}$$

Από αυτή την εξίσωση προκύπτει η βέλτιστη τιμή πιθανότητας p, η οποία χρησιμοποιείται για να επιλεγεί ο βέλτιστος αριθμός cluster-heads. Ως πιθανότητα ορίζεται ο αριθμός cluster-heads που μπορεί να επιλεγεί από τον αριθμό κόμβων στο πεδίο. Επιλέγοντας τη βέλτιστη πιθανότητα, προκύπτει και η βέλτιστη τιμή του συντελεστή ισχύος (power factor) β, ο οποίος είναι ο λόγος μετάδοσης των cluster-heads προς την μετάδοση των κόμβων-μελών. Στην πρώτη επανάληψη οι cluster-heads επιλέγονται τυχαία, εφόσον όλοι οι κόμβοι διατηρούν όλη την αρχική τους ενέργεια και επομένως βρίσκονται σε ομοιογενή κατάσταση. Μετά την πρώτη επανάληψη, για την επιλογή cluster-head για κάθε cluster, χρησιμοποιείται η συνάρτηση κόστους επικοινωνίας, η οποία είναι η εξής:

$$cost_i = f_i + F_i = \frac{(M_K + 1)\sum_{j=1}^{M_k+1} {D_{ij}}^2}{\sum_{k=1}^{M_K+1}\sum_{j=1}^{M_K+1} {D_{kj}}^2} + \frac{(M_K + 1){D_i}^{3.76}}{\sum_{j=1}^{M_K+1} {D_j}^{3.76}}$$

όπου f_i η συνάρτηση κόστους για την intra-επικοινωνία και F_i η συνάρτηση κόστους για την inter-επικοινωνία. Η συνάρτηση κόστους υπολογίζεται για κάθε κόμβο-μέλος σε κάθε cluster και ως επόμενος cluster-head επιλέγεται ο κόμβος με το ελάχιστο κόστος.

Ο *IMPC* συγκρίθηκε με τον *Low Energy Adaptive Clustering Hierarchy (LEACH),* ο οποίος τερματίζει γρήγορα μετά από μερικούς γύρους και δεν εγγυάται καλή κατανομή των cluster-heads, και τον *Energy Aware Multi-Hop Multi-Path Hierarchy (EAMMH).* Τα αποτελέσματα της προσομοίωσης έδειξαν πως ο *IMPC,* αν και είναι κάπως σύνθετος, έχει καλύτερη απόδοση όσον αφορά την κατανάλωση ενέργειας, τη διάρκεια ζωής του δικτύου και τον αριθμό κόμβων που εξαντλούν την ενέργειά τους.

4.3.4 NDCMC: A Hybrid Data Collection Approach for Large-Scale WSNs Using Mobile Element and Hierarchical Clustering

Η συλλογή δεδομένων από ένα μεγάλης κλίμακας WSN είναι μια δύσκολη εργασία και υπάρχουν δύο τρόποι για να αυξηθεί η αποδοτικότητά της: με ιεραρχική δρομολόγηση με βάση την ομαδοποίηση κόμβων και με τη χρήση κινητών στοιχείων (mobile elements – MEs). Δεδομένου ότι και οι δύο μέθοδοι έχουν πλεονεκτήματα και μειονεκτήματα, σε αυτή τη δημοσίευση επιχειρείται ο συνδυασμός τους με μια υβριδική προσέγγιση που ονομάζεται ***Node Density based Clustering and Mobile Collection (NDCMC)*** [59]. Σε αυτή τη μέθοδο, ένας αριθμός cluster-heads (CHs) συγκεντρώνει πληροφορίες από τα μέλη του cluster και ύστερα ένα κινητό στοιχείο επισκέπτεται τους cluster-heads για να συλλέξει τα δεδομένα μέσω single-hop μικρής εμβέλειας ραδιοεπικοινωνίας. Η επιλογή των cluster-heads γίνεται με ένα νέο σχέδιο βασισμένο στην πυκνότητα των κόμβων και η διαδρομή του κινητού στοιχείου σχεδιάζεται από

έναν νέο αλγόριθμο χαμηλής πολυπλοκότητας. Επίσης, παρουσιάζεται το *Random Clustering and Mobile Collection (RCMC)* σχέδιο με το οποίο επιλέγονται τυχαία οι CHs.

Πιο αναλυτικά, η υβριδική μέθοδος αποτελείται από δύο στάδια: την *αρχικοποίηση του δικτύου* (network initialization stage) και τη *συλλογή δεδομένων* (data collection stage). Κατά το πρώτο στάδιο, κάθε κόμβος αναμεταδίδει στους κόμβους που βρίσκονται στην εμβέλειά του ένα "HELLO μήνυμα" που περιλαμβάνει το ID και την GPS τοποθεσία του. Έτσι, κάθε κόμβος μπορεί να γνωρίζει τον αριθμό των γειτονικών του κόμβων και την θέση τους. Κάθε κόμβος στέλνει επίσης την τοποθεσία του στον σταθμό βάσης με τη χρήση μιας γεωγραφικής multi-hop μεθόδου δρομολόγησης (geographic multi-hop routing method – η οποία αναλύεται στην ενότητα III-B της δημοσίευσης). Μετά τη λήψη αυτών των πληροφοριών, ο σταθμός βάσης σχεδιάζει τη διαδρομή συλλογής δεδομένων από τους CHs (III-D) οι οποίοι επιλέγονται με βάση την πυκνότητά τους (III-C). Εκτός από τους CHs, οι κόμβοι που βρίσκονται στην εμβέλεια του κινητού στοιχείου (mobile element – ME), επισημαίνονται ως Virtual Heads (VHs), οι οποίοι στέλνουν επίσης δεδομένα στο ME. Στη συνέχεια, ο σταθμός βάσης (base station – BS) στέλνει στους κόμβους δύο λίστες – μία με τα IDs και τις τοποθεσίες των επιλεγμένων CHs και μία με πληροφορίες για τους VHs. Με αυτό τον τρόπο κάθε κόμβος λαμβάνει το ρόλο του. Εάν δεν είναι ούτε CH ούτε VH, τότε πρόκειται για Normal Node (ND) και αφού υπολογίσει την απόστασή του από τους CHs και VHs, συσχετίζει τον εαυτό του με τον κοντινότερο. Κατά το δεύτερο στάδιο, οι NDs στέλνουν τα δεδομένα τους στον κοντινότερο CH ή VH με τη χρήση της τοπική γεωγραφικής μεθόδου δρμολόγησης (local geographic routing method – III-B) που περιγράφεται στον Αλγόριθμο 1. Το ME κινείται πάνω από τους κόμβους εκπέμποντας αναγνωριστικά σήματα (beacons), τα οποία μόλις λάβει ένας CH ή VH, στέλνει τα συγκεντρωμένα δεδομένα απευθείας στο ME, το οποίο τα μεταφέρει στον BS.

Η *μέθοδος για την επιλογή των CHs* παρουσιάζεται στον Αλγόριθμο 2 και ο σκοπός είναι να επιλεγούν κόμβοι με μεγάλη πυκνότητα, δηλαδή με πολλούς γειτονικούς κόμβους σε μια ακτίνα R. Αυτή η μέθοδος εξασφαλίζει ότι στις περιοχές που οι κόμβοι είναι πυκνά συγκεντρωμένοι θα υπάρχουν CHs τους οποίους το ME θα επισκεφτεί. Έτσι, η διαδικασία συλλογής δεδομένων γίνεται πιο αποδοτική και παράλληλα ελαχιστοποιείται το φορτίο της intra-cluster δρομολόγησης. Επίσης, εξασφαλίζεται ότι για κάθε κόμβο σε έναν cluster θα υπάρχει ένα multi-hop μονοπάτι προς έναν CH ώστε να μην μείνουν δεδομένα που δεν έχουν συλλεγεί. Ακόμα και απομονωμένοι κόμβοι που δεν ανήκουν σε κάποιο cluster, λαμβάνουν τελικά το ρόλο του CH και τους επισκέπτεται το ME για να συλλέξει τα δεδομένα τους. Τέλος, σημαντικό είναι να αναφερθεί ότι ρυθμίζοντας την ακτίνα των clusters (cluster radius) R καθορίζεται και ο αριθμός CHs που επιλέγονται. Αν η ακτίνα R είναι μικρή, θα προκύψουν περισσότεροι CHs σε clusters μικρότερου

μεγέθους και η διαδρομή του ΜΕ γίνεται μεγαλύτερη, αλλά η multi-hop δρομολόγηση από τους NDs στους CHs ή VHs μειώνεται. Αντίθετα, με μεγάλη ακτίνα R, συντομεύει το δρομολόγιο του ΜΕ και το μεγαλύτερο φόρτο επωμίζονται οι κόμβοι. Συνεπώς, μεταβάλλοντας την ακτίνα R προσδιορίζεται και η ισορροπία μεταξύ κατανάλωσης ενέργειας των κόμβων και μήκους διαδρομής του ΜΕ.

Ο *Optimal Track Planning Algorithm* (Αλγόριθμος 3) προσδιορίζει το βέλτιστο δρομολόγιο για το ΜΕ και έχει χαμηλότερη πολυπλοκότητα από τους παραδοσιακούς TSP (Travelling Salesman Problem) αλγορίθμους. Οι προσομοιώσεις με το Monte Carlo απέδειξαν πως αυτός ο αλγόριθμος βρίσκει αποτελεσματικά τη συντομότερη διαδρομή για να επισκεφτεί το ΜΕ ακόμα και περισσότερους από 100 CHs.

Προτείνεται επίσης ένα *Random Clustering and Mobile Collection Scheme (RCMC)* για υβριδική συλλογή δεδομένων, με χαμηλότερη όμως πολυπλοκότητα. Σε αυτό το scheme κάποιοι κόμβοι προεπιλέγονται τυχαία ως CHs. Αφού στείλουν στον BS την τοποθεσία τους, αυτός σχεδιάζει τη διαδρομή του ΜΕ με τον Αλγόριθμο 3. Η διαδικασία είναι ίδια με του NDCMC, με τη διαφορά ότι στον RCMC δεν υπάρχουν VHs. Λόγω της τυχαίας τοποθεσίας των CHs στο δίκτυο, η απόδοσή του σε σχέση με τον NDCMC είναι υποβαθμισμένη. Παρόλα αυτά, χρησιμεύει πρώτον στη σύγκριση με τον NDCMC – και επομένως στην απόδειξη ότι είναι αποτελεσματικός και παρατείνει τη διάρκεια ζωής του δικτύου – και δεύτερον, μπορεί να χρησιμοποιηθεί όταν κάποιοι κόμβοι είναι πιο ισχυροί θέτοντάς τους CHs ή αν προτιμάται κάποιο scheme χαμηλής πολυπλοκότητας.

Όσον αφορά την τελική αξιολόγηση του προτεινόμενου μηχανισμού, αποδείχτηκε ότι είναι πιο οικονομικός ενεργειακά και με πιο εξισορροπημένη κατανάλωση ενέργειας σε σχέση με κινητά στοιχεία σταθερής τροχιάς (fixed ME tracks). Σε σχέση με τον RCMC, όπως προαναφέρθηκε, είναι πολύ οικονομικότερος και μάλιστα παρατείνει τη διάρκεια ζωής του δικτύου κατά 50%. Επίσης, η απόδοση του NDCMC συγκρίθηκε με τους MILP, CSPLI και SST (άλλες τεχνικές με MEs) που προτάθηκαν στο [60] και είχε σημαντικά καλύτερα αποτελέσματα χάρη στην επιλογή των CHs με βάση την πυκνότητα και την χρήση των VHs. Τέλος, σε σύγκριση με αμιγώς βασισμένη-στην-ομαδοποίηση συλλογή δεδομένων, και πιο συγκεκριμένα σε σύγκριση με τον LEACH [55], στον οποίο οι CHs επιλέγονται τυχαία και περιοδικά, ο NDCMC σημείωσε περισσότερο χρόνο λειτουργίας για τον ίδιο αριθμό κόμβων με εναπομείνασα ενέργεια πάνω από ένα κατώφλι.

4.3.5 Service-Aware Clustering: An Energy-Efficient Model for the Internet-of-Things

Η τρέχουσα γενιά αλγορίθμων και πρωτοκόλλων δρομολόγησης ασύρματων αισθητήρων (wireless sensor routing algorithms) έχει σχεδιαστεί βασισμένη σε μια μυωπική προσέγγιση στην οποία θεωρείται πως οι κόμβοι-αισθητήρες έχουν τις ίδιες δυνατότητες ανίχνευσης και επικοινωνίας. Η μυωπική δρομολόγηση δεν είναι κατάλληλη για το IoT. Οι αισθητήρες είναι σχεδιασμένοι για να προσφέρουν διαφορετικές υπηρεσίες, γι' αυτό έχουν και διαφορετικά ενεργειακά μοτίβα. Επομένως, εάν αυτό δε ληφθεί υπόψη, μπορεί να οδηγήσει σε ενεργειακή ανισορροπία και κατ' επέκταση βραχύβια δίκτυα αισθητήρων.

Η παρούσα δημοσίευση επανέρχεται, λοιπόν, στο θέμα της απόδοσης ενέργειας στα δίκτυα αισθητήρων προτείνοντας ένα ***Service-Aware Clustering (SAC)*** [61] πρωτόκολλο δρομολόγησης για τη βελτίωση της διαχείρισης της ενέργειας. Το SAC πρωτόκολλο χρησιμοποιεί έναν συγκεντρωτικό μηχανισμό (centralized mechanism) για την αντιμετώπιση του προβλήματος ενεργειακής ανισορροπίας αφήνοντας τον σταθμό βάσης να επιλέξει cluster heads ανάλογα με την εναπομείνασα ενέργεια των κόμβων, τις ενεργειακές ιδιότητες, τον τύπο των κόμβων και το είδος υπηρεσίας που προσφέρουν, καθώς και τη θέση τους στο δίκτυο.

Για την αξιολόγηση της απόδοσης του SAC πρωτοκόλλου/αλγορίθμου διεξήχθησαν διάφορα επιμέρους πειράματα τα οποία περιγράφονται στη συνέχεια:

- ***Απόδοση SAC πρωτοκόλλου.*** Το συγκεκριμένο πείραμα διεξήχθη για να εκτιμήσει την σκοπιμότητα της χρήσης του SAC μοντέλου σε IoT περιβάλλοντα συγκρίνοντάς το με έναν Breadth-First Search (BFS) και έναν Service-Blind Clustering (SBC) αλγόριθμο. Ο BFS είναι ένας μη ιεραρχικός αλγόριθμος δρομολόγησης που θεωρεί επίπεδη τοπολογία δικτύου (flat network topology), ενώ ο SBC είναι ένας ιεραρχικός αλγόριθμος δρομολόγησης ο οποίος, όμως, βασίζεται μόνο στην ισχύ του σήματος για την επιλογή των clusters και την σχέση μεταξύ cluster-head και κόμβων-μελών, παραβλέποντας την επίγνωση της ενέργειας και των υπηρεσιών. Τα αποτελέσματα φανέρωσαν την αποτελεσματικότητα του SAC αλγορίθμου σε σχέση με τους BFS και SBC από άποψη εναπομείνασας ενέργειας. Επίσης προέκυψε πως ο BFS είχε καλύτερη απόδοση από τον SBC, δηλαδή ένας μη ιεραρχικός αλγόριθμος μπορεί να σημειώσει καλύτερη επίδοση από έναν service-blind αλγόριθμο παρότι εφαρμόστηκαν σε ιεραρχική δρομολόγηση.

- ***Η σκοπιμότητα της επίγνωσης των υπηρεσιών (Service Awareness).*** Σε αυτό το πείραμα ο SAC αλγόριθμος συγκρίθηκε με τους energy-aware LEACH [48] και LEACH-Centralized (LEACH-C) [55], καθώς και με τους πιο πρόσφατους DECSA [62] και MOCRN [63] ώστε να εκτιμηθεί η σκοπιμότητα της επίγνωσης των υπηρεσιών. Ο SAC σημείωσε καλύτερη επίδοση και από τους δύο LEACH αλγορίθμους από άποψη ενεργειακής κατανάλωσης ανά γύρο, αλλά και συνολικά. Επίσης, μέσω της απόδειξης της χειρότερης απόδοσης του LEACH, φανερώνεται και η βελτιστότητα των συγκεντρωτικών (centralized) διαδικασιών δρομολόγησης.
- ***Διάρκεια ζωής του δικτύου αισθητήρων: FND και LND.*** Η διάρκεια ζωής του δικτύου αξιολογήθηκε με βάση τον πρώτο και τον τελευταίο κόμβο που εξαντλεί την ενέργειά του – first node to die (FND) και last node to die (LND), αντίστοιχα. Ο σταθμός βάσης τοποθετήθηκε πρώτα στο κέντρο του δικτύου. Οι FND προέκυψαν στον 332 γύρο για τον DECSA, στον 391 για τον MOCRN και στον 448 για τον SAC. Οι LND διατήρησαν την ενέργειά τους μέχρι τους γύρους 728, 831 και 968 για τους DECSA, MOCRN και SAC, αντίστοιχα. Η διαφορά γίνεται μεγαλύτερη όταν ο σταθμός βάσης τοποθετήθηκε έξω από την περιοχή του δικτύου. Τότε οι FND για τους DECSA, MOCRN και SAC προέκυψαν στους γύρους 255, 298 και 361, ενώ οι LND στους 556, 746 και 802, αντίστοιχα. Τα αποτελέσματα αυτά φαίνονται και στον πίνακα 10.

BS at the Center	**DECSA**	**MOCRN**	**SAC**
FND (round)	332	391	448
LND (round)	728	831	968
BS Outside	**DECSA**	**MOCRN**	**SAC**
FND (round)	255	298	361
LND (round)	556	746	802

5. Διάρκεια ζωής του δικτύου: FND και LND

Ο SAC, λοιπόν, τρέχει για περισσότερους γύρους προτού ο πρώτος και ο τελευταίος κόμβος εξαντλήσουν την ενέργειά τους. Ο λόγος είναι ότι έλαβε υπόψη παραμέτρους όπως την τρέχουσα ενέργεια των κόμβων-αισθητήρων, τη διαφοροποίηση των υπηρεσιών και την απόσταση μεταξύ κόμβων κατά την επιλογή νέων cluster-heads.

- ***Διάρκεια ζωής του δικτύου αισθητήρων: Μέση διάρκεια ζωής και επεκτασιμότητα.*** Η διάρκεια ζωής του δικτύου μπορεί να οριστεί ως το άθροισμα την αρχικής ενέργειας όλων των κόμβων διαιρούμενο με τη συνολική ενέργεια που χάθηκε ανά κόμβο. Η μέση διάρκεια ζωής που πέτυχε ο SAC για 100 κόμβους είναι 10.8% υψηλότερη από του DECSA και 2.4%

από του MOCRN. Στο πείραμα με τους 500 κόμβους η μέση διάρκεια ζωής του SAC προέκυψε 23.7% υψηλότερη από του DECSA και 5.3% από του MOCRN. Τα καλά αποτελέσματα του SAC οφείλονται στην εύκολη επικοινωνία των κόμβων με τους cluster-heads οι οποίοι επιλέχτηκαν με βάση την τοποθεσία και την εναπομείνασα ενέργεια των κόμβων, ενώ στον DECSA μόνο με βάση την εναπομείνασα ενέργεια και στον MOCRN με βάση την απόσταση. Επίσης, ο SAC σχηματίζει λιγότερους clusters και αυτό συνεπάγεται ότι είναι περισσότερο επεκτάσιμος.

- ***Απόδοση διαχείρισης κίνησης (traffic engineering performance).*** Το τελευταίο πείραμα αξιολόγησε την διαχείριση της κίνησης εξετάζοντας την φειδώ με την οποία χρησιμοποιήθηκε η ενέργεια. Τα αποτελέσματα φανέρωσαν την ενεργειακή λιτότητα του νέου – βασισμένου στο LIBP – πρωτοκόλλου σε σχέση με τα CTP και RPL.

4.3.6 Density-based Energy-efficient Clustering Algorithm for Wireless Sensor Networks

Ο ***Density-based Energy-efficient Clustering Algorithm (DECA)*** [64] είναι ακόμα ένας αλγόριθμος που στοχεύει στη μείωση της κατανάλωσης ενέργειας σε ένα WSN. Στα πλαίσιο αυτής της προσπάθειας, για την επιλογή των cluster-heads, εκτός από την εναπομείνασα ενέργεια των κόμβων, ο DECA λαμβάνει υπόψη του και την πυκνότητά τους. Έτσι, εγγυάται ομοιόμορφη κατανομή των cluster-heads. Επίσης, σχεδιάστηκαν νέοι αλγόριθμοι intra-cluster και multi-hop inter-cluster δρομολόγησης που εξοικονομούν ενέργεια σε κάποιο βαθμό.

Πιο αναλυτικά, για την επιλογή των cluster-heads, σε κάθε γύρο κατατάσσονται σε φθίνουσα σειρά οι τιμές πυκνότητας όλων των κόμβων και επιλέγεται ο πρώτος, δηλαδή εκείνος με την μεγαλύτερη πυκνότητα. Επειδή οι γειτονικοί τους κόμβοι είναι λογικό να έχουν παρόμοια πυκνότητα, η οποία είναι αρκετά μεγάλη ώστε να παρεμποδίσει την επιλογή στους επόμενους γύρους, αποκλείονται όλοι οι κόμβοι που εντοπίστηκαν στους προηγούμενους γύρους. Για να αποφευχθεί η εξάντληση της ενέργειας των cluster-heads, ορίζεται ο λόγος n της εναπομείνασας ενέργειας προς την αρχική ενέργεια. Όταν ο λόγος n ενός cluster-head γίνει μικρότερος από ένα κατώφλι 10%, τότε σταματά να είναι υπεύθυνος για την συνάθροιση των δεδομένων και επιλέγεται νέος. Στην διαδικασία δρομολόγησης του DECA κάυε κόμβος στέλνει δεδομένα στον cluster-head απευθείας, μέσω ενός βήματος (hop). Ο αντίστοιχος cluster-head για κάθε κόμβο είναι εκείνος που απέχει μικρότερη απόσταση. Η inter-routing επικοινωνία, σε αντίθεση με του LEACH [48], γίνεται με multi-hop τρόπο. Για κάθε cluster-head επιλέγεται ο βέλτιστος

relay cluster-head (επόμενος κατά τη δρομολόγηση cluster-head) ο οποίος διατηρεί τη μικρότερη ενεργειακή κατανάλωση. Η κατανάλωση αυτής της διαδρομής συγκρίνεται με την απευθείας αποστολή των δεδομένων από τον εκάστοτε cluster head προς το σταθμό βάσης και τελικά καθορίζεται η βέλτιστη inter-routing με βάση την μικρότερη σπατάλη ενέργειας.

Όπως ήταν αναμενόμενο, η κατανομή των cluster-heads προέκυψε πολύ πιο ομοιόμορφη από εκείνη του LEACH στο ίδιο WSN. Λόγω του multi-hop τρόπου inter-cluster δρομολόγησης, η συνολική κατανάλωση ενέργειας μειώνεται όσο οι clusters αυξάνονται. Βέβαια, το ποσοστό μείωσης γίνεται σχετικά μικρό αφότου κάποιοι clusters έχουν σχηματιστεί, επειδή οι περισσότεροι κόμβοι συμπεριλαμβάνονται σε κάποιον υπάρχοντα cluster με μικρή απόσταση μετάδοσης ως τον cluster-head. Σε σύγκριση με τον LEACH, αλλά και τον Density-based Clustering Protocol (DBCP) που αποτελεί βελτίωση του LEACH, επιβεβαιώθηκε η καλύτερη ενεργειακή απόδοση του DECA. Τέλος, όσον αφορά την διάρκεια ζωής του δικτύου, στον LEACH ο πρώτος κόμβος εξάντλησε την ενέργειά του στον 94 γύρο, ο DBCP στον 124 και ο DECA στον 248 – που είναι διπλάσιος αριθμός γύρων από την περίπτωση του DBCP.

4.4 Στόχος: Μείωση πολυπλοκότητας

4.4.1 Fast Modified Global k-means Algorithm for Incremental Cluster Construction

Ο k-means και οι παραλλαγές του είναι γενικά γρήγοροι αλγόριθμοι. Είναι, όμως, ευαίσθητοι στην επιλογή των σημείων εκκίνησης (starting points) – δηλαδή τα αρχικά κέντρα των clusters – και ανεπαρκείς για προβλήματα ομαδοποίησης σε μεγάλα σύνολα δεδομένων. Τα τελευταία χρόνια, έχουν γίνει προσπάθειες να ξεπεραστούν οι δυσκολίες με την επιλογή των σημείων εκκίνησης, μέσω σταδιακών προσεγγίσεων (incremental approaches). Τέτοιοι αλγόριθμοι είναι ο *Global k-means (GKM)* και ο *Modified Global k-means (MGKM)*, οι οποίοι προσθέτουν ένα cluster center σε κάθε επανάληψη. Αυτή η προσέγγιση, ενώ βελτιώνει τον k-means, χρειάζεται να αποθηκεύει ολόκληρη την «μήτρα συγγένειας» (affinity matrix) ή να την υπολογίζει σε κάθε επανάληψη. Αυτό είναι χρονοβόρο και απαιτεί αρκετή μνήμη, ακόμα και για μετρίου μεγέθους σύνολα δεδομένων.

Οι συγγραφείς, λοιπόν, στην παρούσα δημοσίευση [65], προτείνουν τον ***Fast Modified Global k-means (FMGKM)*** αλγόριθμο, μια νέα εκδοχή του MGKM αλγορίθμου, στην οποία χρησιμοποιείται μία “auxiliary cluster function” που παράγει ένα σύνολο σημείων εκκίνησης σε διαφορετικά σημεία του συνόλου δεδομένων. Ο k-means εφαρμόζεται εκκινώντας από αυτά τα σημεία ώστε να ελαχιστοποιήσει την

βοηθητική αυτή συνάρτηση (auxiliary cluster function) και η καλύτερη λύση επιλέγεται ως το σημείο εκκίνησης για το επόμενο cluster center. Εκμεταλλεύονται, δηλαδή, πληροφορίες που έχουν συλλεχθεί σε προηγούμενες επαναλήψεις του incremental αλγορίθμου. Με αυτό τον τρόπο εξαλείφεται η ανάγκη υπολογισμού ή αποθήκευσης ολόκληρης της μήτρας συγγένειας (affinity matrix) και ως αποτέλεσμα μειώνεται η υπολογιστική προσπάθεια (computational effort) και η χρήση μνήμης (memory usage). Σε αυτό το σημείο σημειώνεται πως έχει ήδη προταθεί μια γρήγορη εκδοχή του MGKM [66], με στόχο να ελαττωθεί η υπολογιστική του πολυπλοκότητα. Στόχος, όμως, της δημοσίευσης [65] είναι κυρίως, όπως προαναφέρθηκε, ο περιορισμός της χρήσης της μνήμης. Επίσης, στο [66] εξακολουθεί να χρειάζεται ο υπολογισμός μέρους της μήτρας συγγένειας, ενώ στο [65] όχι.

Πιο συγκεκριμένα, λοιπόν, στον *FMGKM* η υπολογιστική προσπάθεια μειώνεται με τους εξής 2 τρόπους:

1. Αφαιρώντας σημεία δεδομένων που είναι κοντά στα κέντρα των clusters που βρέθηκαν στην προηγούμενη επανάληψη, δηλαδή το $(k-1)$-partition. Με αυτό τον τρόπο αποκλείουμε σημεία δεδομένων από (α) τη λίστα των σημείων που μπορούν να προσελκύσουν μεγάλους clusters και (β) τη λίστα των σημείων που μπορεί να προσελκύονται από σημεία δεδομένων που δεν έχουν αποκλειστεί. Αυτό μας επιτρέπει να μειώσουμε σημαντικά τον αριθμό των υποψήφιων σημείων εκκίνησης και τον αριθμό των data points που μπορεί να προσελκύονται από έναν υποψήφιο. Εν ολίγοις, εκμεταλλευόμενοι την σταδιακή (incremental) φύση του αλγορίθμου, αποφεύγουμε τον υπολογισμό της μήτρας συγγένειας. Ολόκληρο το σύνολο δεδομένων χρησιμοποιείται μόνο στην πρώτη επανάληψη, όταν υπολογίζεται το κεντροειδές (centroid) του συνόλου δεδομένων.
2. Χρησιμοποιώντας την τριγωνική ανισότητα (triangle inequality) για αποστάσεις, ώστε να αποφευχθούν περιττοί υπολογισμοί. Απαραίτητο είναι να αποθηκευτεί στη μνήμη η μήτρα που περιέχει αποστάσεις μεταξύ σημείων δεδομένων και κέντρων των clusters από την $(k-1)$-οστή επανάληψη. Αυτή η μήτρα, όμως, είναι πολύ μικρότερη από την μήτρα συγγένειας.

Όσον αφορά την υπολογιστική πολυπλοκότητα, ο *FMGKM* χρειάζεται λιγότερη υπολογιστική προσπάθεια από τον GKM και τον MGKM. Με τον αλγόριθμο στο [66] έχουν παρόμοια υπολογιστική πολυπλοκότητα.

Τα αποτελέσματα των αριθμητικών πειραμάτων αποδεικνύουν ότι ο προτεινόμενος αλγόριθμος είναι, στις περισσότερες περιπτώσεις, γρηγορότερος και ακριβέστερος από τον GKM. Συγκρινόμενος με τον MGKM, δίνει παρόμοια αποτελέσματα, απαιτεί όμως λιγότερη αξιολόγηση βάσει νόρμας (less norm evaluations) και χρόνο επεξεργασίας (CPU time). Η βελτίωση που προσφέρει ο *Fast Modified Global k-means (FMGKM)* γίνεται όλο και πιο ουσιαστική όσο το μέγεθος του συνόλου δεδομένων αυξάνει.

4.4.2 An agglomerative clustering algorithm using a dynamic k-nearest-neighbor list

Ο νέος αλγόριθμος [67] στοχεύει στην μείωση της υπολογιστικής πολυπλοκότητας της μεθόδου του Ward (γνωστή και ως ιεραρχική συσσωρευτική ομαδοποίηση – hierarchical agglomerative clustering) η οποία είναι $O(N^3)$ [68]. Ο *Double Linked Algorithm (DLA)* κάνει χρήση ενός προσεγγιστικού γράφου k-κοντινότερων γειτόνων για συσσωρευτική ομαδοποίηση [69] και μπορεί να μειώσει σημαντικά τον χρόνο υπολογισμού της μεθόδου Ward με την εύρεση μιας προσεγγιστικής λύσης. Μεταξύ άλλων, ένας ακόμα αλγόριθμος που έχει προταθεί για τη μείωση του χρόνου υπολογισμού της μεθόδου Ward, είναι ο *Fast Pairwise Nearest Neighbor (FPNN)* [70] με υπολογιστική πολυπλοκότητα $O(\tau N^2)$, όπου τ ο μέσος αριθμός clusters προς ενημέρωση σε κάθε στάδιο συγχώνευσης clusters.

Ο νέος αλγόριθμος ***Dynamic k-Nearest-Neighbor (DKNNA)*** [67] στοχεύει στη λύση του προβλήματος ότι ο DLA βρίσκει μόνο μια προσεγγιστική λύση συσσωρευτικής ομαδοποίησης, διατηρώντας παράλληλα τη χαμηλή υπολογιστική πολυπλοκότητα του DLA. Η προτεινόμενη μέθοδος χρησιμοποιεί μια δυναμική KNN (k-nearest-neighbor) λίστα στην οποία αποθηκεύονται οι k κοντινότεροι γείτονες για κάθε cluster. Η KNN λίστα για κάθε σημείο δεδομένων πρέπει να ορίζεται κατά τη διαδικασία αρχικοποίησης (initialization process), όπως και στον DLA. Γι' αυτό, ύστερα από κάθε διαδικασία συγχώνευσης, η προτεινόμενη μέθοδος προσδιορίζει πρώτα ένα σύνολο clusters των οποίων οι κοντινότεροι γείτονες πρέπει να ενημερωθούν. Στη διαδικασία συγχώνευσης και ενημέρωσης κάθε επανάληψης, ενημερώνονται οι KNN λίστες των clusters που επηρεάζονται από την διαδικασία συγχώνευσης. Εάν οι KNN λίστες είναι κενές για κάποιους από τους clusters που ενημερώνονται, οι κοντινότεροί τους γείτονες προσδιορίζονται ερευνώντας όλους τους clusters. Επομένως, η προτεινόμενη μέθοδος εγγυάται την ακρίβεια των κοντινότερων γειτόνων ενός cluster και μπορεί να εξασφαλίσει ένα εξίσου καλό αποτέλεσμα με τον FPNN και τη μέθοδο του Ward. Σημείο κλειδί αυτής της μεθόδου είναι ότι η διαδικασία ενημέρωσης αποφεύγεται μέχρι η KNN λίστα του cluster που ενημερώνεται να αδειάσει.

Από τα πειράματα που διεξήχθησαν, φαίνεται πως ο DKNNA επιτυγχάνει τα ίδια σχεδόν αποτελέσματα ομαδοποίησης με τον FPNN. Συγκρινόμενος με τον FPNN με γρήγορη αναζήτηση (Fast Search – FS) για την εύρεση των κοντινότερων γειτόνων (FPNN + FS), η προτεινόμενη μέθοδος σε συνδυασμό με τον ίδιο αλγόριθμο γρήγορης αναζήτησης (DKNNA + FS) μειώνει τον χρόνο υπολογισμού κατά 1.90-2.18 για το σύνολο δεδομένων από μια πραγματική εικόνα και κατά 1.92-2.02 χρησιμοποιώντας το σύνολο δεδομένων που παράγεται από τρεις εικόνες. Επίσης, ο DKNNA + FS μπορεί να ελαττώσει το μέσο τετραγωνικό σφάλμα κατά 1.26% για το ίδιο σύνολο δεδομένων.

4.4.3 An efficient hyperellipsoidal clustering algorithm for resource-constrained environments

Σε πολλές περιπτώσεις, όπως στα ασύρματα δίκτυα αισθητήρων, όπου οι υπολογιστικές δυνατότητες είναι περιορισμένες, χρειάζονται αλγόριθμοι ομαδοποίησης που να είναι όσο το δυνατόν λιγότερο υπολογιστικά ακριβοί. Η συγκεκριμένη δημοσίευση εστιάζει σε αυτό το πρόβλημα και προτείνει έναν ισχυρό αλγόριθμο με χαμηλή υπολογιστική πολυπλοκότητα, κατάλληλο για περιβάλλοντα με υπολογιστικούς περιορισμούς. Ένας καλός αλγόριθμος ομαδοποίησης θα έπρεπε να έχει μικρό αριθμό παραμέτρων εισόδου και να είναι επαρκώς «αναίσθητος» σε αλλαγές αυτών των παραμέτρων. Στον προτεινόμενο αλγόριθμο, που ονομάζεται ***Hyperellipsoidal clustering for resource-constrained environments (HyCARCE)*** [71], μόνο μία παράμετρος πρέπει να ρυθμιστεί από τον χρήστη: το αρχικό μέγεθος των κελιών του πλέγματος (grid cell size). Τα κύρια χαρακτηριστικά του HyCARCE είναι τα εξής:

- Αυτόματη επιλογή του αριθμού των clusters
- Χαμηλό υπολογιστικό κόστος $O(N)$
- Ρητή ανίχνευση ορίων κάθε cluster
- Ενσωματωμένη ανίχνευση outlier

Σκοπός του αλγορίθμου είναι να βρει ένα σύνολο clusters $C = \{c_j : j = 1 \dots K\}$ σε ένα σύνολο δεδομένων N καταγραφών $S = \{s_k : k = 1 \dots N\}$ όπου κάθε καταγραφή $s_k \in R^d$ είναι ένα διάνυσμα d-διαστάσεων. Κάθε cluster c_j αντιπροσωπεύεται από ένα υπερελλειψοειδές e_j που σηματοδοτεί το όριο του cluster. Η γενική μορφή ενός υπερελλειψοειδούς ορίου ορίζεται από ένα σύνολο σημείων X στον R^d που ικανοποιούν την εξίσωση $(X - m)^T A(X - m) = 1$, όπου m είναι το κέντρο του ελλειψοειδούς και A ένας $d \times d$ συμμετρικός θετικά ορισμένος πίνακας που ονομάζεται χαρακτηριστικός πίνακας, του οποίου τα ιδιοδιανύσματα προσδιορίζουν τις κύριες κατευθύνσεις του υπερελλειψοειδούς και το αντίστροφο

της τετραγωνικής ρίζας των ιδιοτιμών είναι οι αντίστοιχες ισημερινές ακτίνες. Ο λόγος που επιλέχθηκαν ελλειψοειδή για την αναπαράσταση των clusters είναι ότι έχουν την ευελιξία να μοντελοποιούν από σφαιρικές έως γραμμικές κατανομές δεδομένων και μπορούν επίσης να παραμετροποιηθούν από το κέντρο και τον χαρακτηριστικό τους πίνακα.

Τα κύρια βήματα του HyCARCE είναι τα παρακάτω:

1. *Αρχικοποίηση:*

   Ο χώρος εισόδου χωρίζεται σε σταθερού μεγέθους κελιά d-διαστάσεων όπως στους grid-based αλγορίθμους και εντοπίζονται τα μη κενά κελιά. Στο τέλος αυτού του βήματος, υπάρχει ένα σύνολο κελιών που το καθένα περιέχει τουλάχιστον ένα σημείο δεδομένων.

2. *Περικοπή κελιών:*

   Τα κελιά με τυπική απόκλιση μικρότερη του μέσου αριθμού σημείων δεδομένων σε ένα κελί, αφαιρούνται. Σε κάθε κελί, προσαρμόζεται ένα υπερελλειψοειδές πάνω στα δεδομένα έτσι ώστε να καλύπτει τουλάχιστον το 95% αυτών. Αυτά τα ελλειψοειδή αποτελούν το σημείο εκκίνησης της διαδικασίας επέκτασης.

3. *Επέκταση και αναπροσαρμογή:*

   Τα ελλειψοειδή μεγεθύνονται και προσαρμόζουν τα όριά τους ώστε να «φιλοξενήσουν» τα νέα σημεία δεδομένων. Κάθε βήμα μεγέθυνσης ακολουθείται από ένα βήμα αναπροσαρμογής, στο οποίο ενημερώνεται το κέντρο του ελλειψοειδούς, σύμφωνα με τα νέα σημεία που προστέθηκαν. Χρησιμοποιώντας αυτό το κέντρο ως δειγματική μέση τιμή (sample mean), υπολογίζεται η καινούρια μήτρα συνδιασποράς (covariance matrix), για τα δεδομένα του μεγεθυμένου υπερελλειψοειδούς, εντός ενός κατωφλίου. Η διαδικασία αυτή συνεχίζεται έως ότου η μεταβολή των σημείων δεδομένων εντός του υπερελλειψοειδούς γίνει μικρότερη από ένα κατώφλι σ, το οποίο ορίζεται ως $\sigma = \frac{1}{19} n_v$ όπου n_v ο αριθμός των σημείων δεδομένων εντός του ελλειψοειδούς πριν την μεγέθυνση. Αυτό είναι, λοιπόν, το κριτήριο τερματισμού.

4. *Αφαίρεση πλεοναζόντων ελλειψοειδών:*

   Είναι πιθανό να υπάρχουν πολλαπλά ελλειψοειδή σε κάθε cluster, γι' αυτό το τελευταίο βήμα είναι να απομακρυνθούν τα πλεονάζοντα ελλειψοειδή. Αν η απόσταση μεταξύ των κέντρων δυο ελλειψοειδών είναι μικρότερη από ένα κατώφλι M_t, τότε αφαιρείται εκείνο με τα λιγότερα σημεία δεδομένων. Τα ελλειψοειδή που παραμένουν, σηματοδοτούν τα όρια των clusters. Τέλος, αν ένα σημείο δεδομένων «πέφτει» μέσα σε περισσότερα από ένα ελλειψοειδή, τότε

χρησιμοποιείται η ελάχιστη Mahalanobis απόσταση του σημείου από τα κέντρα τους για να επιλεγεί ο καταλληλότερος για αυτό cluster.

Η πολυπλοκότητα του αλγορίθμου είναι κοντά στο $O(N)$ σε σχέση με τον αριθμό των σημείων δεδομένων.

Για την αξιολόγηση του αλγορίθμου, ο HyCARCE συγκρίθηκε με άλλους γνωστούς αλγορίθμους ως προς την ακρίβεια και το υπολογιστικό κόστος, σε διαφόρων ειδών σύνολα δεδομένων. Ο DENCLUE, ο k-means και ο *Gustafson-Kessel (GK)* [72] έχουν συγκρίσιμη υπολογιστική πολυπλοκότητα με τον HyCARCE, ενώ ο *Subtractive Clustering (SC)* [73] είναι υπολογιστικά ακριβότερος. Ο μεν *SC* χρησιμοποιήθηκε στη σύγκριση διότι είναι παρόμοιος με τον HyCARCE ως προς τον τρόπο με τον οποίο βρίσκει το κέντρο των clusters και το όριο απόφασης (decision boundary) για την ομαδοποίηση των δεδομένων – ο δε *GK* διότι είναι ένας προηγμένος ασαφής (fuzzy) αλγόριθμος ομαδοποίησης και χρησιμοποιείται ως σημείο αναφοράς για την ακρίβεια η οποία αποδείχτηκε συγκρίσιμη ή καλύτερη.

Επίσης, αξιολογήθηκε η ευαισθησία στις παραμέτρους εισόδου και ο χρόνος εκτέλεσης του HyCARCE σε σχέση με του DENCLUE ο οποίος θεωρείται εξαιρετικά ακριβής. Ο HyCARCE αποδείχτηκε λιγότερο ευαίσθητος στην επιλογή παραμέτρων από τον DENCLUE, επομένως επιτυγχάνει πιο ισχυρά αποτελέσματα για ένα ευρύ φάσμα ρυθμίσεων των παραμέτρων. Παρόλα αυτά, η ισχύς των αποτελεσμάτων επηρεάζεται αρνητικά σε υψηλότερες διαστάσεις, γι' αυτό και είναι καταλληλότερος για δεδομένα χαμηλών διαστάσεων. Ως προς το χρόνο εκτέλεσης, ο HyCARCE αποδείχτηκε σημαντικά ταχύτερος από τον DENCLUE και τον GK, για όλα τα σύνολα δεδομένων. Ο HyCARCE χρειάστηκε κατά μέσο όρο χρόνο εκτέλεσης (CPU time) 0.28 δευτερόλεπτα, ενώ ο DENCLUE 11 δευτερόλεπτα, το οποίο είναι περίπου δύο τάξεις μεγέθους μεγαλύτερο.

4.4.4 A new topological clustering algorithm for interval data

Οι περισσότεροι αλγόριθμοι ομαδοποίησης ορίζονται για την αντιμετώπιση διανυσματικών δεδομένων (vectorial data) στον R^d. Αυτό το είδος αναπαράστασης χρησιμοποιείται συχνά για να αναλύσει δεδομένα από φυσικές μετρήσεις, όμως υπάρχουν πολλά άλλα είδη πληροφοριών που δε μπορούν να περιγραφούν με διανύσματα. Τέτοια σύνθετα δεδομένα περιγράφονται για παράδειγμα με ένα κείμενο, μια εικόνα ή μια ιεραρχική δομή. Αυτό το άρθρο εστιάζει στα συνεχή δεδομένα ή δεδομένα διαστήματος (interval data). Τα interval data είναι αριθμητικά, αλλά από τη φύση τους, δεν έχουν σημείο μηδενισμού – δηλαδή σημείο στο οποίο δεν υπάρχει καθόλου η μετρούμενη ποσότητα, όπως για παράδειγμα η

θερμοκρασία. Στο διανυσματικό χώρο, τα interval data ορίζονται από υπερ-ορθογώνια (hyper-rectangles) και μοντελοποιούν ποσότητες που μεταβάλλονται μεταξύ δύο ορίων.

Οι συγγραφείς παρουσιάζουν έναν νέο αλγόριθμο ομαδοποίησης για interval data [74], βασισμένο στη μάθηση ενός *Self-Organizing Map (SOM)* [75]. Ο SOM είναι μια δημοφιλής, μη γραμμική τεχνική για τη μείωση των διαστάσεων (dimensionality) και την οπτικοποίηση δεδομένων, με πολύ χαμηλό υπολογιστικό κόστος. Άλλοι αλγόριθμοι που έχουν προταθεί σχετικά με SOMs, θεωρούνται εργαλεία για διανυσματικό κβαντισμό (vector quantization) και οπτικοποίηση interval δεδομένων και δε μπορούν να χρησιμοποιηθούν απευθείας για να ομαδοποιήσουν τα δεδομένα. Ο προτεινόμενος αλγόριθμος του ***"A new topological clustering algorithm for interval data"*** [74] είναι μια μέθοδος ομαδοποίησης δύο επιπέδων για interval data. Η ομαδοποίηση δύο επιπέδων βασισμένη σε SOM, σε πρώτο επίπεδο συνδυάζει τη μείωση διαστάσεων (dimension reduction) και τις δυνατότητες ταχείας εκμάθησης του SOM για την κατασκευή ενός μειωμένου χώρου και σε δεύτερο επίπεδο εφαρμόζει μια μέθοδο ομαδοποίησης σε αυτόν τον νέο χώρο για την παραγωγή των τελικών clusters. Οι μέθοδοι δύο επιπέδων μειώνουν τον υπολογιστικό χρόνο και επιτρέπουν μια οπτική ερμηνεία των αποτελεσμάτων ομαδοποίησης. Ένα μειονέκτημά τους είναι ότι η τμηματοποίηση των δεδομένων με χρήση του SOM δεν είναι η βέλτιστη καθώς μέρος της πληροφορίας χάνεται κατά το πρώτο στάδιο που είναι η μείωση των διαστάσεων. Επιπλέον, ο διαχωρισμός σε δύο στάδια δεν είναι κατάλληλος για σταδιακή (incremental) τμηματοποίηση δεδομένων που κινούνται στο χρόνο. Για τη λύση αυτών των προβλημάτων, έχει προταθεί ο *S2L-SOM (Simultaneous Two-Levels-SOM)* αλγόριθμος [76], ο οποίος εκτελεί ταυτόχρονη μάθηση και ομαδοποίηση του SOM από πληροφορίες των δεδομένων. Ο νέος αλγόριθμος είναι μια επέκταση του S2L-SOM. Το σημαντικότερο πλεονέκτημά του σε σύγκριση με τις υπάρχουσες μεθόδους είναι πως ο αριθμός των clusters προσδιορίζεται αυτόματα, χωρίς να χρειάζεται a priori υπόθεση για τον αριθμό των clusters που απαιτούνται. Αυτό επιτυγχάνεται εφόσον στο τέλος της διαδικασίας ομαδοποίησης, κάθε cluster είναι ένα σύνολο προτύπων που συνδέονται μεταξύ τους με γειτονικές συνδέσεις. Έτσι, ο αριθμός των clusters προσδιορίζεται αυτόματα.

Μέσω πειραμάτων, αποδείχτηκε η αποτελεσματικότητα του αλγορίθμου στη λύση διαφόρων ειδών προβλημάτων ομαδοποίησης. Συγκρινόμενος με υπάρχουσες μεθόδους ομαδοποίησης interval δεδομένων (*DIV, SCLUST* και τις τέσσερις ιεραρχικές μεθόδους που περιέχονται στον *SHICLUST*) ως προς την ποιότητα, αποδείχτηκε καλύτερος από τις DIV και SCLUST μεθόδους και παρόμοιος με την SHICLUST. Παρόλα αυτά, ο προτεινόμενος αλγόριθμος επιτυγχάνει αυτή την ποιότητα με γραμμική πολυπλοκότητα, κάτι που δεν ισχύει για τον SHICLUST. Επομένως, η προτεινόμενη μέθοδος προσφέρει άριστα

αποτελέσματα σε μικρό χρόνο επεξεργασίας. Σύγκριση έγινε και με τις μεθόδους SYKSOM και SYKCLUST. Ο SYKSOM δεν καταφέρνει να τον ανταγωνιστεί, ενώ τα αποτελέσματα είναι παρόμοια με του SYKCLUST, στον οποίο όμως απαιτείται ο προσδιορισμός του αριθμού των clusters εκ των προτέρων. Συνοψίζοντας, τα κύρια πλεονεκτήματα του προτεινόμενου αλγορίθμου είναι:

- Ο αριθμός των clusters προσδιορίζεται αυτόματα
- Ταξινομεί ομάδες μη-κυρτής μορφής
- Η διαδικασία είναι αξιόπιστη και γρήγορη
- Η πολυπλοκότητα είναι γραμμική του αριθμού των δεδομένων

4.4.5 A time-efficient pattern reduction algorithm for k-means clustering

Οι περισσότεροι αλγόριθμοι ομαδοποίησης, έρχονται πλέον αντιμέτωποι με πολύ μεγάλα σύνολα δεδομένων που απαιτούν online επεξεργασία. Αν ληφθεί υπόψη και η παράμετρος της ποιότητας που πρέπει να επιτευχθεί, ο χρόνος απόκρισης είναι μείζονος σημασίας. Καθώς ο χρόνος υπολογισμού για ομαδοποίηση δεδομένων υψηλών διαστάσεων είναι ανάλογος με τον αριθμό των διαστάσεων των δεδομένων εισόδου, κάποιοι εστίασαν την έρευνά τους στη μείωση των διαστάσεων των δεδομένων εισόδου, ελαττώνοντας τον αριθμό των χαρακτηριστικών των αρχικών δεδομένων.

Αυτό το άρθρο παρουσιάζει τον ***Pattern Reduction (PR)*** αλγόριθμο [77], ο οποίος λειτουργεί σε συνδυασμό με τον k-means και άλλους βασισμένους σε αυτόν αλγορίθμους, με σκοπό τη μείωση του χρόνου υπολογισμού. Ως κίνητρο υπήρξε η παρατήρηση πως οι περισσότεροι βασισμένοι στον k-means αλγόριθμοι συμπεριφέρονται εξαιρετικά όμοια σε μεταγενέστερο στάδιο της σύγκλισης, με την έννοια ότι οι περισσότεροι υπολογισμοί επαναλαμβάνονται, χωρίς όμως να συμβάλλουν στην τελική λύση. Γι' αυτό, αν υπολογισμοί του k-means (όπως ο υπολογισμός του κέντρου, των αποστάσεων με τα κέντρα και την ανάθεση μοτίβων στο κοντινότερο κέντρο) που δεν χρειάζεται να επαναληφθούν και είναι στην ουσία περιττοί, μπορούν να εξαλειφθούν προκειμένου να εξοικονομηθεί χρόνος υπολογισμού. Σε κάθε επανάληψη, επιλέγονται μοτίβα (patterns) τα οποία δεν αλλάζουν στο εξής τη συμμετοχή (membership) τους, οπότε συμπιέζονται και απομακρύνονται. Μοτίβα που συμπιέζονται και απομακρύνονται, παραμένουν στον cluster από τον οποίο αφαιρέθηκαν. Σε επόμενες επαναλήψεις, ένα ήδη συμπιεσμένο μοτίβο μπορεί να συμπιεστεί ξανά και να απομακρυνθεί. Πολύ σημαντικός είναι ο προσδιορισμός της κατάλληλης στιγμής για την έναρξη της λειτουργίας του PR. Αν μοτίβα που δεν αλλάζουν τη συμμετοχή τους σε επόμενες επαναλήψεις συμπιεστούν και αφαιρεθούν στις πρώτες επαναλήψεις, τότε μπορεί να

εξοικονομηθεί σημαντική ποσότητα χρόνου υπολογισμού διατηρώντας την ποιότητα του αποτελέσματος. Αν, όμως, συμπιεστούν και απομακρυνθούν πολύ νωρίς, τότε ο k-means μπορεί να «πέσει» σε τοπικό ελάχιστο και να συγκλίνει σε ανακριβές αποτέλεσμα. Εάν πάλι, αυτό συμβεί πολύ αργά, δε θα επιτευχθεί μείωση του χρόνου υπολογισμού. Εξίσου σημαντικός είναι και ο ορισμός του ορίου απομάκρυνσης (removal bound). Εάν οριστεί σε μεγαλύτερη τιμή, επιτρέπει την αφαίρεση περισσότερων μοτίβων και έτσι περισσότερος χρόνος υπολογισμού θα εξοικονομηθεί. Ωστόσο, κάτι τέτοιο μπορεί να επιφέρει μεγαλύτερη απώλεια ποιότητας. Από πειραματικά αποτελέσματα προέκυψε πως ρυθμίζοντας το removal bound στο 80% αποφέρει το καλύτερο αποτέλεσμα, παρέχοντας μια καλή ισορροπία μεταξύ του χρόνου υπολογισμού και του ποσοστού της ακρίβειας.

Αρχικά, ο συνδυασμένος αλγόριθμος k-means με patter reduction, λειτουργεί όπως ακριβώς ο k-means, εκτός από το ότι συνεχίζει να ελέγχει αν είναι η σωστή στιγμή για να ξεκινήσει ο PR αλγόριθμος. Αν καθορίσει αυτήν την κατάλληλη στιγμή και το removal bound δεν έχει επιτευχθεί, τότε ο PR εφαρμόζεται, και συνεχίζει ο έλεγχος για την παύση του PR με βάση το removal bound. Στην ουσία, η λειτουργία του PR αλγορίθμου χωρίζεται σε δύο μέρη: 1) την συμπίεση του μοτίβου και αφαίρεση (*pattern compression and removal – PCR*) και 2) την αντιστοίχιση μοτίβου και την ανανέωση μέσου (*pattern assignment and mean update – PAMU*). Η διαδικασία του PCR δείχνει πώς συμπιέζονται και αφαιρούνται τα μοτίβα. Αρχικά, ο PR απαιτεί τον ορισμό ενός removal bound το οποίο υποδηλώνει το ποσοστό των μοτίβων που επιτρέπεται να συμπιεστούν και να αφαιρεθούν. Αυτό το όριο τίθεται λόγω την ανάγκης να μειωθεί ο βαθμός με τον οποίο ο θόρυβος επηρεάζει τα αποτελέσματα ομαδοποίησης. Πειραματικά αποτελέσματα δείχνουν πως η ρύθμιση του removal bound στο 80% δίνει ικανοποιητικά αποτελέσματα. Αν το ποσοστό αυτό οριστεί πολύ μεγάλο μπορεί να μειώσει το ποσοστό ακρίβειας, ενώ αν οριστεί πολύ μικρό τότε επιβάλλει ένα όριο στο ποσό του χρόνου υπολογισμού που μπορεί να μειωθεί. Αν το removal bound δεν έχει επιτευχθεί, τότε ο PCR ψάχνει τα μοτίβα που βρίσκονται κοντά στο μέσο του cluster και επομένως μπορούν να αφαιρεθούν. Όπως ο PCR, έτσι και ο PAMU εκτελείται εφόσον το removal bound δεν έχει επιτευχθεί. Ο PAMU απαιτεί τη σύγκριση των αποστάσεων μεταξύ κάθε μοτίβου και όλων των μέσων ώστε να προσδιοριστεί ο cluster στον οποίο ανήκει κάθε μοτίβο. Επίσης, αντιστοιχίζει μοτίβα που δεν συμπιέστηκαν και αφαιρέθηκαν στους clusters που ανήκουν και ύστερα υπολογίζει το νέο μέσο του κάθε cluster.

Ο PR εφαρμόστηκε στους εξής 5 κλασικούς αλγορίθμους ομαδοποίησης: k-means (KM), Relational k-means (RKM), Kernel k-means (KKM), Triangle inequality k-means (TKM) και Genetic k-means (GKA) ώστε να αξιολογηθεί η απόδοσή του. Ο PR μείωσε τον χρόνο υπολογισμού των KM, RKM, TKM, KKM και GKA

κατά 79%, 51%, 81%, 82% και 74%, αντίστοιχα. Αξίζει να σημειωθεί ότι ο RKM μπορεί να μειώσει τον χρόνο υπολογισμού του KM κατά τουλάχιστον κατά 84% και ο TKM κατά 64%, όμως ο PR μπορεί να κάνει περαιτέρω μείωση κατά 51% και 81%, αντίστοιχα. Εν συντομία, αποδείχτηκε ότι με μια μικρή απώλεια στην ποιότητα, ο PR μπορεί να βελτιώσει τον χρόνο υπολογισμού όλων των παραπάνω αλγορίθμων.

4.4.6 Fuzzy joint points based clustering algorithms for large data sets

Η Fuzzy joint points (FJP) [78] μέθοδος είναι μία από τις επιτυχημένες ασαφείς (fuzzy) προσεγγίσεις για την density-based ομαδοποίηση. Με βάση την FJP έχουν αναπτυχθεί και άλλες μέθοδοι όπως η Noise-Robust FJP (NRFJP) και η Fuzzy Neighborhood DBSCAN (FN-DBSCAN). Αυτές οι μέθοδοι αν και έχουν αξιοσημείωτα πλεονεκτήματα σε σχέση με τον DBSCAN, έχουν ως μειονέκτημα την χαμηλή ταχύτητα, γι΄ αυτό δε μπορούν να χρησιμοποιηθούν σε εφαρμογές με μεγάλες βάσεις δεδομένων. Η Modified FJP (MFJP) μέθοδος αντιμετωπίζει αυτό το πρόβλημα και βελτιώνει ως έναν βαθμό την ταχύτητα, αλλά δεν είναι τόσο ικανοποιητική από άποψη εφαρμογής. Η χρονική πολυπλοκότητα των FJP και MFJP αλγορίθμων είναι $O(n^4)$ και $O(n^3 \log_2 n)$, αντίστοιχα. Παρά την βελτίωση στην MFJP μέθοδο, η $O(n^3 \log_2 n)$ εξακολουθεί να είναι υψηλή πολυπλοκότητα για τη διαδικασία της ομαδοποίησης λαμβάνοντας υπόψη τα υπερμεγέθη σύνολα δεδομένων που υπάρχουν σε όλες σχεδόν τις εφαρμογές. Τα δύο “bottlenecks” αυτών των μεθόδων είναι:

- Ο υπολογισμός της transitive closure matrix
- Ο προσδιορισμός της κρίσιμης (critical) τιμής του «α» (alpha)

Στην παρούσα δημοσίευση, επιχειρείται η ενσωμάτωση κάποιων μεθόδων στους βασισμένους στον FJP αλγορίθμους, ώστε να βελτιωθούν περαιτέρω από την άποψη της χρονικής πολυπλοκότητας χειρότερης περίπτωσης και του χρόνου τρεξίματος και να γίνει εφικτή η χρήση τους σε μεγαλύτερα σύνολα δεδομένων. Επίσης, προτείνεται ένας ακόμα γρηγορότερος αλγόριθμος ο οποίος χρησιμοποιεί την FJP προσέγγιση με σχετικά επιβλεπόμενο (supervised) τρόπο.

Για τον υπολογισμό της transitive closure matrix οι ερευνητές δανείζονται τον βέλτιστο αλγόριθμο του [79], ο οποίος μπορεί να υλοποιηθεί με δύο τρόπους και ανάλογα τον τρόπο ονομάζεται «διαδικασία TC» ή «διαδικασία TC-heap». Η χρονική πολυπλοκότητα της TC διαδικασίας είναι $O(n^2 \log_2 n)$, ενώ της TC-heap $O(n^2)$, γι’ αυτό και προτιμάται η δεύτερη. Με τη χρήση, λοιπόν, της TC-heap διαδικασίας για τον υπολογισμό της max-min transitive closure, η πολυπλοκότητα του MFJP αλγορίθμου πέφτει σε

$O(n^2 \log_2 n)$. Όσον αφορά το «κρίσιμο α» (η μέγιστη διαφορά - the maximum gap), η τιμή του επηρεάζει τα χωρίσματα της ομαδοποίησης (clustering partitions). Οι τιμές του «α» αντιστοιχούν στις unique values στην transitive closure matrix η οποία έχει τα στοιχεία της ταξινομημένα κατά φθίνουσα σειρά. Παρόλα αυτά, η ταξινόμηση του πίνακα δεν είναι απαραίτητη για την εύρεση του maximum gap. Η διαδικασία MaxGap βρίσκει το maximum gap σε έναν μη ταξινομημένο πίνακα και επιστρέφει την κρίσιμη τιμή του α, α_z. Η MaxGap τρέχει σε γραμμικό χρόνο, επομένως χρησιμοποιώντας την εξαλείφεται η $O(n^2 \log_2 n)$ πολυπλοκότητα της διαδικασίας ταξινόμησης, για την εύρεση του α_z.

Η αντικατάσταση των χρονοβόρων βημάτων του MFJP με τις προαναφερθείσες διαδικασίες οδηγεί σε έναν αλγόριθμο βέλτιστου χρόνου που τρέχει σε $O(n^2)$ χρόνο και ονομάζεται ***Optimal FJP (OFJP)*** [80]. Ο ψευδοκώδικας του OFJP δίνεται στην εικόνα 11.

OFJP(X):
Input: Data set $X = \{x_1, x_2, \ldots, x_n\}$.
Output: Clustering partition $X^1, X^2, \ldots, X^k$.

Step1. Calculate $d_{ij} = d(x_i, x_j)$; $d_{\max} = \max d_{ij}, i, j = 1, \ldots, n$;
Step2. Calculate the fuzzy neighborhood relation T: $t_{ij} = 1 - \frac{d_{ij}}{d_{\max}}, i, j = 1, \ldots, n$;
Step3. Call the procedure ***TC-heap***(T) to obtain the transitive closure $\hat{T}$;
Step4. Call the procedure ***MaxGap***(V) to obtain the critical alpha value α_z;
Step5. Call the procedure ***Clusters***(X, α_z) to obtain the resulting clustering partition $X^1, X^2, \ldots, X^k$;
End.

6. Optimal FJP (OFJP) αλγόριθμος

Οι FJP, MFJP και OFJP εξαλείφουν το μειονέκτημα του DBSCAN – την μεγάλη του ευαισθησία στις παραμέτρους εισόδου ε και $MinPts$, με τον οποίο μπορούν να επιτευχθούν επιθυμητά αποτελέσματα μόνο για μικρά διαστήματα τιμών αυτών των παραμέτρων. Στην παρούσα δημοσίευση προτείνεται και ένας νέος αλγόριθμος, ο ***αScan (alpha-scan)*** [80], ο οποίος αντί να υπολογίζει την transitive closure matrix για την εξαγωγή της τιμής α_z από αυτήν, σαρώνει την σχέση ασαφούς γειτονιάς (fuzzy neighborhood relation) με διαφορετικές τιμές α_l για την εύρεση μιας κατάλληλης α_z-γειτονιάς. Για κάθε α_l-γειτονιά, οι συνδεδεμένες συνιστώσες (components) της fuzzy neighborhood relation μήτρας πρέπει να ανακαλυφθούν. Κάθε στοιχείο αντιστοιχεί σε έναν cluster. Τα χωρίσματα ομαδοποίησης θα είναι ίδια για κάποιες διαφορετικές τιμές του α_l και η μέγιστη απόσταση μεταξύ των clusters (maximum inter-cluster distance) επιτυγχάνεται εκεί που ένα χώρισμα (partition) επαναλαμβάνεται περισσότερο. Σε αντίθεση με τις αυτόνομες FJP μεθόδους, ο αScan έχει μια παράμετρο εισόδου $\Delta\alpha$, η οποία είναι η

μονάδα σάρωσης. Η χρονική πολυπλοκότητα του νέου αυτού αλγορίθμου είναι $O(rn^2)$, όπου r είναι ο μέγιστος αριθμός φορών που επαναλαμβάνεται ο βρόχος μεταξύ βημάτων 3 και 5 που φαίνονται στος ψευδοκώδικα του αScan στην εικόνα 12. Η πολυπλοκότητα της διαδικασίας $ConnectedComponents$ (εικόνα 13) είναι $O(n^2)$ εφόσον υπολογίζει την γειτονιά για κάθε σημείο δεδομένων. Ο αριθμός των επαναλήψεων r θα είναι μεγάλος για μικρές τιμές του $\Delta\alpha$ και αντίστροφα. Επομένως, επιλέγοντας σχετικά μεγάλη τιμή για την παράμετρο $\Delta\alpha$, επιτυγχάνεται μείωση του χρόνου εκτέλεσης. Αν και η πολυπλοκότητα του αScan είναι μεγαλύτερη από του OFJP, η τιμή του r μπορεί να θεωρηθεί αμελητέα στην πράξη. Ανάλογα με την τιμή της παραμέτρου $\Delta\alpha$, ο αScan μπορεί να τρέξει σε συντομότερο χρόνο από τον OFJP.

αScan(X, $\Delta\alpha$):
Input: Data set $X = \{x_1, x_2, \ldots, x_n\}$ and parameter $\Delta\alpha$.
Output: Clustering partition $X^1, X^2, \ldots, X^c$.

Step1. Calculate $d_{ij} = d(x_i, x_j)$; $d_{\max} = \max d_{ij}, i, j = 1, \ldots, n$;
$R = \emptyset$; $c = r = s = 0$; $\alpha_l = 1 - \Delta\alpha$;
/* α_l: α value of the current partition
c: number of clusters of the last different partition
r: number of repetitions of the current partition
s: number of repetitions of the most repeated partition */
Step2. Calculate the fuzzy neighborhood relation $T : t_{ij} = 1 - \frac{d_{ij}}{d_{\max}}, i, j = 1, \ldots, n$;
Step3. Call the procedure ***ConnectedComponents(X, T, α_l)*** to obtain a clustering partition $X^1, X^2, \ldots, X^k$.
Step4. If $k = c$ then, $r = r + 1$; /* note that k is the number of clusters obtained at Step3 */
Else,
If $s < r$ and $k \neq 1$, then
$R = \{X^1, X^2, \ldots, X^k\}$; $s = r$; $c = k$; $\alpha_z = \alpha_l + \Delta\alpha$;
End if;
$r = 1$;
End if;
Step5. $\alpha_l = \alpha_l - \Delta\alpha$;
If $k > 1$ and $\alpha_l > 0$ then, go to Step3;
Else,
the resulting clustering partition is R, the number of clusters is c and the critical alpha value is α_z;
End if;
End.

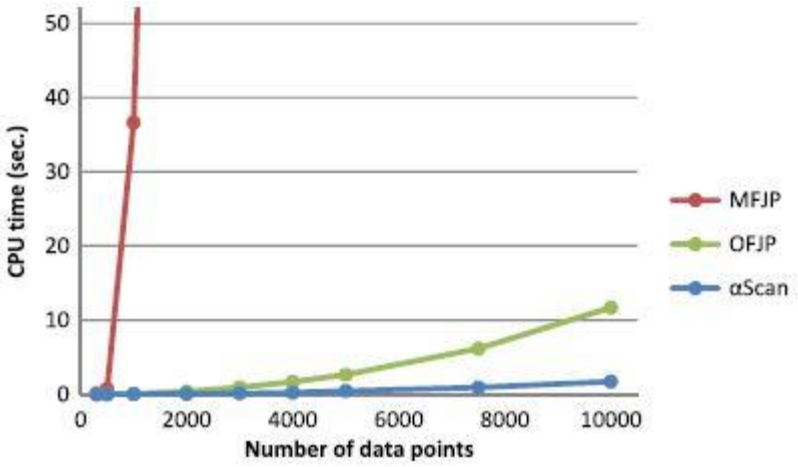
7. Αλγόριθμος αScan

Procedure ConnectedComponents (X, T, α_l):
Input: Data set X, fuzzy neighborhood relation T and parameter α_l.
Output: Clustering partition $X^1, X^2, \ldots, X^k$

Step1. $S = F(X)$, where X is the global data set; $k = 1$;
Step2. Pick an element $C \in S$; $N = \{C\}$; $X^k = \emptyset$;
Step3. While $N \neq \emptyset$, do
pick an element $A \in N$ to form $X^k = X^k \cup \{B \in S | T(A, B) \geq \alpha\}$;
$N = X^k \backslash \{A\}$; $S = S \backslash \{A\}$;
End while;
Step4. If $S \neq \emptyset$, then
$k = k + 1$; go to Step2;
Else,
return the partition $X^1, X^2, \ldots, X^k$ and k;
End if;
End.

8. *Διαδικασία ConnectedComponents*

Οι FJP, MFJP και OFJP είναι ίδιοι από την άποψη απόδοσης, οπότε αξιολογήθηκαν με βάση την αποδοτικότητα του χρόνου τρεξίματος. Τα αποτελέσματα έδειξαν πως ο OFJP είναι δραματικά γρηγορότερος από τον MFJP. Αυτή η επιτάχυνση του αλγορίθμου, κάνει τον FJP κατάλληλο για μεγάλα σύνολα δεδομένων. Περαιτέρω βελτίωση της ταχύτητας επετεύχθη με τη χρήση του αScan, ο οποίος είναι προτιμητέος όταν η ταχύτητα αποτελεί το κύριο μέλημα. Η οπτικοποίηση των αποτελεσμάτων γίνεται στο παρακάτω γράφημα:



9. Χρόνοι τρεξίματος των αλγορίθμων vs. αριθμός σημείων δεδομένων

4.5 Στόχος: Βελτίωση ποιότητας

4.5.1 A Cloud-Friendly RFID Trajectory Clustering Algorithm in Uncertain Environments

Στα πλαίσια ανάπτυξης του Διαδικτύου των Πραγμάτων, μέσω διασύνδεσης δισεκατομμυρίων ετικετών ταυτοποίησης μέσω ραδιοσυχνοτήτων (Radio Frequency Identification tags - RFID tags) και αισθητήρων στο διαδίκτυο, θα παράγεται ένας πρωτοφανής αριθμός συναλλαγών και ποσότητα δεδομένων που απαιτούν μια νέα προσέγγιση εξόρυξης χρήσιμων πληροφοριών από RFID τροχιές (trajectories). Η ταυτοποίηση μέσω ραδιοσυχνοτήτων έχει την ικανότητα να εξάγει αυτόματα και ασύρματα πληροφορίες από μικροηλεκτρονικές ετικέτες που είναι κολλημένες πάνω σε αντικείμενα, μέσω ραδιοκυμάτων. Σε ένα δίκτυο αναγνώρισης ραδιοσυχνοτήτων συνδέονται απομονωμένα συστήματα RFID με άλλο λογισμικό και έτσι μπορούμε να έχουμε εφαρμογές ιχνηλασιμότητας (traceability applications). Παρόλα αυτά, τα RFID δεδομένα συνήθως περιέχουν ένα σημαντικό βαθμό αβεβαιότητας ως προς την αξιοπιστία τους, που προκαλείται από διάφορους παράγοντες, όπως ελαττώματα στο hardware, σφάλματα μετάδοσης και περιβαλλοντική αστάθεια. Κάποιες φορές μπορεί οι ετικέτες να μην διαβαστούν καθόλου, πράγμα που μας οδηγεί στο λανθασμένο συμπέρασμα ότι το συγκεκριμένο αντικείμενο δεν είναι παρόν (αυτό ονομάζεται “missing reading”). Έτσι, οι τροχιές των αντικειμένων γίνονται ελλιπείς και επηρεάζεται η παρακολούθηση και ο εντοπισμός μεμονωμένων αντικειμένων (όπως πχ. ενός δέματος που ταξιδεύει), αλλά προκύπτουν και ανακριβή στατιστικά στοιχεία που οδηγούν σε μεροληπτικές (biased) επιχειρηματικές αποφάσεις, όπως σε περιπτώσεις διαχείρισης εφοδιαστικής αλυσίδας, που ο σχεδιασμός της διαδρομής διανομής παίζει σημαντικό ρόλο για τον έλεγχο του κόστους.

Μια προσέγγιση για τη λύση αυτού του προβλήματος, θα ήταν ο σχεδιασμός πιο σύνθετων ετικετών, λιγότερο ευάλωτων σε φαινόμενα θωράκισης (shielding) και παρεμβολής (interference). Όμως, για να εξαπλωθεί παντού η RFID τεχνολογία, πρέπει να είναι και οικονομικά προσιτή. Είναι επομένως πιο εφικτό να γίνει μέσω ενός στρώματος λογισμικού (software layer). Έχουν υπάρξει πολλές “data cleaning” και “probabilistic event extraction” μέθοδοι (ενδεικτικά [81], [82], [83], [84]) ως ενδιάμεσο λογισμικό (middleware) για να βελτιώσουν την ποιότητα των RFID δεδομένων – οι περισσότερες όμως δεν έχουν ως στόχο τις RFID τροχιές, ειδικά όταν πρόκειται για διεσπαρμένη εφαρμογή (distributed application) που αναπτύσσεται σε ευρεία περιοχή.

Στο [85], λοιπόν, προτείνεται μια προσέγγιση του προβλήματος μέσω λογισμικού, το οποίο μπορεί να χειριστεί αυτές τις αβεβαιότητες σε διεσπαρμένες εφαρμογές ιχνηλασιμότητας μέσω RFID. Εν συντομία, η συνεισφορά τους έχει ως εξής:

- Προτείνουν ένα νέο μέτρο ομοιότητας (similarity measure) σχεδιασμένο για RFID τροχιές και το οποίο μπορεί να μεταχειριστεί variants σε διαστάσεις χρόνου και χώρου.
- Προτείνουν έναν αποδοτικό αλγόριθμο ομαδοποίησης που είναι πολύ λιγότερο ευαίσθητος σε θόρυβο και outliers από τις υπάρχουσες μεθόδους. Ο αλγόριθμος παραλληλίστηκε στο προγραμματιστικό μοντέλο MapReduce ώστε να κάνει καλύτερη χρήση των πόρων της υπολογιστικής νέφους (cloud computing).

Ως outlier, στο παρόν πρόβλημα, ορίζεται η τροχιά στην οποία υπάρχουν “missing readings”. Ζητούμενο είναι να ανακτηθούν. Υποθέτοντας ότι οι περισσότερες τροχιές αντικειμένων έχουν καταγραφεί σωστά, και ότι έχουμε τα πλήρη trajectory clusters SC (δηλαδή μια ακολουθία ζευγών κόμβων που περιγράφει τη χωροχρονική σχέση μιας ομάδας αντικειμένων), η διαδικασία ανάκτησης μπορεί να μετατραπεί σε πρόβλημα ταξινόμησης (classification). Τις περισσότερες φορές όμως το σύνολο των trajectory clusters δεν είναι γνωστό εκ των προτέρων ή μπορεί να αλλάζει ενίοτε. Ως αποτέλεσμα, είναι απαραίτητο να παραχθεί το SC ομαδοποιώντας τις υπάρχουσες τροχιές. Αν ομαδοποιηθούν οι τροχιές σε διαφορετικές κλάσεις, τότε είναι δυνατό να εντοπιστούν outliers και να ανακτηθούν τα missing readings. Επομένως, η ομαδοποίηση είναι το ζήτημα κλειδί.

Ένα ερώτημα που προκύπτει είναι τι μέτρο ομοιότητας (similarity measure) θα χρησιμοποιηθεί για να μετρηθεί η ομοιότητα μεταξύ RFID τροχιών. Επειδή μεταβάλλεται και η χρονική εκτός από την χωρική διάσταση, δεν μπορούν να χρησιμοποιηθούν απλά μοντέλα όπως η Ευκλείδεια απόσταση, η Dynamic Time Warping (DTW), η Edit Distance with Real Penalty (ERP), η Longest Common Subsequence (LCSS) και η Edit Distance on Real Sequences (EDR).

Το προτεινόμενο μοντέλο μέτρησης ομοιότητας *Time-Parameterized Edit Distance (TED)* προκύπτει από το EDR εάν αντικατασταθεί η παράμετρος “subcost” από ένα χρονικά παραμετροποιημένο κόστος (time-parameterized cost) και είναι το εξής:

$$TED(TR_1, TR_2) = \begin{cases} |TR_1|, & \alpha\nu\ |TR_2| = 0 \\ |TR_2|, & \alpha\nu\ |TR_1| = 0 \\ \min\begin{cases} dist_t\big(TR_1(1), TR_2(1)\big) + TED\big(Rest(TR_1), Rest(TR_2)\big), \\ TED(Rest(TR_1), TR_2) + 1, TED\big(TR_1, Rest(TR_2)\big) + 1 \end{cases}, & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

όπου εισάγεται και η *Time-parameterized Distance*, $dist_t(e_1, e_2)$, για δύο στοιχεία e_1 και e_2 σε δύο τροχιές. Επίσης, προτείνεται ένα "recall-oriented" μοντέλο μέτρησης ομοιότητας που ονομάζεται *Time-parameterized Longest Common Subsequences (TLCSS)* για να αντιμετωπιστούν διαφορετικές απαιτήσεις ανάλογα την εργασία εξόρυξης.

Ο αλγόριθμος ομαδοποίησης που προτείνεται *(Algorithm 1:* ***Hierarchical RFID Trajectory Clustering - HRTC****)* [85] είναι ένας ιεραρχικός αλγόριθμος. Η ιδέα είναι να ομαδοποιηθούν πρώτα τα σημεία που προβάλλονται στο (T_s, T_e) επίπεδο, όπου T_s είναι το χρονικό διάστημα των arrival readings και T_e το χρονικό διάστημα των leaving readings. Ύστερα, κάθε cluster σε αυτό το επίπεδο, επεκτείνεται στην τρίτη διάσταση, την τοποθεσία. Κατά την επέκταση αυτή , οι clusters «σπάνε» σε υπό-ομάδες (sub-clusters) που παριστάνουν τα κλαδιά εκείνου του κόμβου.

Είσοδος του αλγορίθμου είναι το σύνολο των τροχιών και έξοδος το σύνολο των ομάδων τροχιών (trajectory clusters), καθώς φυσικά και το σύνολο των outliers. Πρώτα, για όλους τους υπάρχοντες clusters, χωρίζονται όλες οι τροχιές που ανήκουν σε αυτά, σε διαφορετικά σύνολα, σύμφωνα με την επόμενη στάση τους. Ύστερα, για κάθε σύνολο, χρησιμοποιείται ο αλγόριθμος $OPTICS_t$ στις χρονικές διαστάσεις της επόμενης στάσης, ο οποίος είναι ικανός να ανακτήσει missing readings και αυτή η διαδικασία συνεχίζεται αναδρομικά.

Ο παραπάνω αλγόριθμος μπορεί να χειριστεί τα outliers σε διαστάσεις χρόνου, αλλά δεν μπορεί να χειριστεί τα missing readings. Αυτό συμβαίνει γιατί ένας ιεραρχικός αλγόριθμος θα χειριστεί διαφορετικά ένα μονοπάτι (path) $p_1 = \{v_1, v_2, v_3, v_4\}$ από το μονοπάτι $p'_1 = \{v_1, v_3, v_4\}$ στο οποίο υπήρξε ένα missing reading, το v_2. Αντίθετα, με την παρατήρηση ότι το p'_1 είναι υπομονοπάτι του p_1, μπορούν να συγχωνευτούν αυτοί οι δύο clusters και να ανακτηθούν τα missing readings. Αυτό κάνει ο δεύτερος αλγόριθμος που προτείνεται σε αυτή τη δημοσίευση *(Algorithm 2: Trajectory Cluster Merging)*. Αφού ταξινομήσει τα clusters με βάση τα μήκη των μονοπατιών τους, ξεκινά από τα μικρότερα μονοπάτια, βρίσκει όλα τα υποψήφια clusters και διαλέγει προς συγχώνευση εκείνο με τη μικρότερη TED απόσταση και που ικανοποιεί κάποιο δεδομένο κατώφλι.

Στην πειραματική αξιολόγηση, ο *Hierarchical RFID Trajectory Clustering (HRTC)* αλγόριθμος συγκρίθηκε ως προς την ποιότητα ομαδοποίησης με τους Time-Focused Clustering (TFC) [86] και Fuzzy C-Means (FCM) αλγορίθμους. Ως μέτρο ποιότητας χρησιμοποιήθηκε το Άθροισμα του Τετραγωνικού Σφάλματος (Sum of Squared Error – SSE) και αποδείχτηκε ότι ο *HRTC* έχει υψηλότερες επιδόσεις από τους άλλους δύο όταν ο αριθμός των clusters γίνεται αρκετά μεγάλος – 70 σε αυτό το πείραμα. Οι αλγόριθμοι

συγκρίθηκαν και ως προς την αποδοτικότητα της ομαδοποίησης από άποψη χρόνου. Εδώ ο TFC ήταν καλύτερος, επειδή δεν δίνει βάση στις αβεβαιότητες των δεδομένων οπότε δεν σπαταλά επιπλέον χρόνο σε αυτές. Από τους δύο αλγορίθμους που υπολογίζουν τις αβεβαιότητες, όμως, ο HRTC ήταν καλύτερος από τον FCM, αφού ο πρώτος χωρίζει τα δεδομένα σε μέρη και ύστερα τρέχει τον αλγόριθμο σε κάθε partition, ενώ ο δεύτερος χρειάζεται να τρέξει αρκετές φορές σε όλες τις τροχιές του συνόλου. Παρόλα αυτά, όταν το επίπεδο αβεβαιότητας δεν είναι υψηλό, ο HRTC και ο TFC έχουν παρόμοιο χρόνο τρεξίματος.

Όσον αφορά τα μέτρα ομοιότητας, όπως ήταν αναμενόμενο, τα προτεινόμενα TED και TLCSS παρήγαγαν καλύτερης ποιότητας ομαδοποίηση από ό,τι τα LCSS και EDR. Αυτό οφείλεται κυρίως στο γεγονός ότι λαμβάνουν υπόψη τη διάσταση του χρόνου.

4.5.2 A New Data Clustering Algorithm

Στο [87] οι συγγραφείς προτείνουν έναν νέο αλγόριθμο, τον ***Data Clustering Algorithm Combining Induction and Deduction (IDC),*** ο οποίος συνδυάζει την *Επαγωγική μέθοδο (Inductive method)* με την *Παραγωγική μέθοδο (Deductive method)*. Όλοι οι κλασικοί αλγόριθμοι ομαδοποίησης χρησιμοποιούν τη μέθοδο της επαγωγής, δηλαδή αναλύουν γενικές θεωρίες βασιζόμενοι πάνω σε συγκεκριμένα δεδομένα, εφόσον ομαδοποιούν κατευθείαν αντικείμενα δεδομένων σε clusters σύμφωνα με τις ομοιότητές τους. Μέσω αυτής της μεθοδολογίας υπάρχει μια αβεβαιότητα όσον αφορά τα αποτελέσματα που προκύπτουν. Ένα παράδειγμα είναι ότι εφόσον ισχύει $2^{2^1}+1=5$, $2^{2^2}+1=17$, $2^{2^3}+1=257$, $2^{2^4}+1=65537$, και είναι όλοι πρώτοι αριθμοί, θα υποθέταμε ότι όλοι οι αριθμοί της μορφής $2^{2^n}+1 (n \in N^*)$ είναι πρώτοι. Για $n=5$, η υπόθεση αυτή, όμως, δεν είναι σωστή, εφόσον $2^{2^5}+1=4294967297=641*6700417$. Είναι σκόπιμος, λοιπόν, ο συνδυασμός επαγωγικής και παραγωγικής μεθόδου, όπως στο παράδειγμα, ώστε να βελτιωθεί το μονόπλευρο αρχικό συμπέρασμα.

Η ιδέα αυτή έχει εφαρμοστεί ξανά, αλλά όχι σε ομαδοποίηση. Κάποιοι άλλοι ερευνητές εφάρμοσαν την ομαδοποίηση δύο φορές, όπως στον αλγόριθμο BIRCH [7], προκειμένου να βελτιωθεί η ποιότητα, αλλά αυτό δε συμβαίνει πάντα εφόσον δουλεύουν πάνω στο ήδη ομαδοποιημένο σύνολο δεδομένων.

Ο *IDC*, λοιπόν, χωρίζει το σύνολο δεδομένων σε *k* αρχικά γκρουπ, στα οποία τα σημεία δεδομένων δεν ανήκουν σε κάποιον συγκεκριμένο cluster. Τα περιττά γκρουπ εφαρμόζουν την επαγωγική μέθοδο, ενώ τα άρτια γκρουπ εφαρμόζουν την παραγωγική μέθοδο στο επαγωγικό αποτέλεσμα του προηγούμενου

γκρουπ και ύστερα συγχωνεύονται. Το συγχωνευμένο γκρουπ στην επόμενη επανάληψη λειτουργεί είτε επαγωγικά είτε παραγωγικά. Μετά από κάθε επανάληψη, ο αριθμός των γκρουπ υποδιπλασιάζεται, και ύστερα από $\log k$ επαναλήψεις μένει μόνο ένα γκρουπ, του οποίου το αποτέλεσμα οδηγεί το αποτέλεσμα ομαδοποίησης ολόκληρου του συνόλου δεδομένων. Στην i-οστή $(i = 1, \dots, [\log k])$ επανάληψη υπάρχουν $k/2^{i-1}$ ζεύγη επαγωγικών και παραγωγικών γκρουπ, δηλαδή $k/2^{i-1}$ διαδικασίες μέσα σε μια επανάληψη.

Η Επαγωγική Διαδικασία (The Inductive Process)

Κατά τη διάρκεια κάθε επανάληψης, τα περιττά γκρουπ λειτουργούν επαγωγικά, συνοψίζουν τα αποτελέσματα και εφοδιάζουν με το συμπέρασμα το διπλανό παραγωγικό γκρουπ. Στην πρώτη επανάληψη, ομαδοποιούνται τα αρχικά αντικείμενα δεδομένων στα επαγωγικά γκρουπ και μετά τα αντικείμενα σε αυτά τα γκρουπ είναι το σύνολο των clusters και τα «προσωρινά» (provisional) outliers (ονομάζονται προσωρινά, γιατί ύστερα από περαιτέρω ομαδοποίηση μπορεί να μην είναι πια outliers). Έτσι, στην επαγωγική διαδικασία, ομαδοποιούνται τα σύνολα των clusters και τα προσωρινά outliers, και συνοψίζονται τα χαρακτηριστικά κάθε cluster μετά την ομαδοποίηση. Η ομαδοποίηση ξεκινά αφού έχουν προστεθεί τα αντικείμενα από την προηγούμενη επαγωγή, και τα χαρακτηριστικά των clusters ετοιμάζονται για την επόμενη επαγωγική διαδικασία.

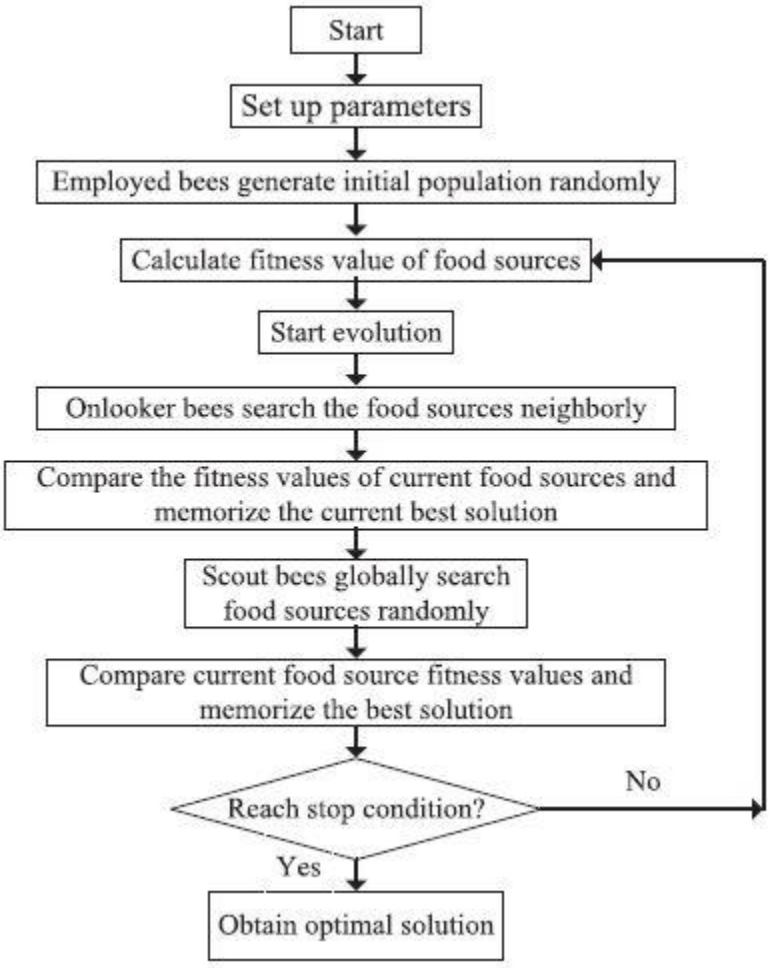
Η Παραγωγική Διαδικασία (The Deductive Process)

Σκοπός της παραγωγικής διαδικασίας είναι επαληθεύσει τα συμπεράσματα που προκύπτουν από την επαγωγική διαδικασία. Τα παραγωγικά αντικείμενα (deductive objects) περιλαμβάνουν αρχικά 1) πρωτότυπα (original) αντικείμενα δεδομένων και 2) clusters που προέκυψαν από ομαδοποίηση. Τα πρωτότυπα αντικείμενα δεδομένων ελέγχονται για το αν ταιριάζουν με τα χαρακτηριστικά του cluster που προέκυψε απ' την επαγωγική διαδικασία και έτσι κρίνεται αν ανήκουν σε κάποιον cluster – αν όχι, ορίζονται ως προσωρινά outliers. Για τα clusters, εφόσον έχουμε τα χαρακτηριστικά τους, αποφασίζεται εάν θα συγχωνευτούν ή όχι, με βάση κάποιο κατώφλι επικάλυψης των χαρακτηριστικών των clusters.

Η χρονική πολυπλοκότητα του IDC είναι $O(n^2)$. Στα πειράματα, συγκρίθηκε με τον BIRCH και τον *k*-means και αποδείχτηκε ότι έχει μεγαλύτερη ακρίβεια και ικανότητα να ανιχνεύει outliers. Χρειάζεται όμως να βελτιωθεί ώστε να μπορεί να χειριστεί δεδομένα υψηλών διαστάσεων.

4.5.3 Automatic kernel clustering with bee colony optimization

Ένα σημαντικό μειονέκτημα πολλών μεθόδων ομαδοποίησης, είναι ότι απαιτούν από τον χρήστη τον αριθμό των clusters προκαταβολικά. Δυστυχώς, στα περισσότερα προβλήματα του πραγματικού κόσμου, δεν υπάρχουν διαθέσιμες πληροφορίες όσον αφορά τον αριθμό τους εκ των προτέρων. Όμως, ο αριθμός τους επηρεάζει σημαντικά το αποτέλεσμα της ομαδοποίησης. Γι' αυτό, μια πολλά υποσχόμενη λύση σε αυτό το πρόβλημα είναι οι αυτόματοι αλγόριθμοι ομαδοποίησης, οι οποίοι προσδιορίζουν αυτόματα τον αριθμό clusters και τους κατασκευάζουν. Ένας τέτοιος αλγόριθμος είναι ο προτεινόμενος ***Automatic kernel clustering with bee colony optimization (AKC-BCO)*** [88], ο οποίος χρησιμοποιεί μια ευρετική (heuristic) μέθοδο – τον bee colony optimization (BCO) αλγόριθμο που αναλύεται στο [89] – και μια συνάρτηση πυρήνα (kernel function) που αντιμετωπίζει τα προβλήματα των μη-γραμμικά διαχωρίσιμων δεδομένων. Το διάγραμμα ροής του AKC-BCO, με βάση τη λογική του BCO, είναι το παρακάτω:



15. Διάγραμμα ροής του AKC-BCO

Ως μέτρο εγγύτητας, αντί της συμβατικής Ευκλείδειας απόστασης, χρησιμοποιείται μια συνάρτηση πυρήνα (kernel function), με κριτήριο το CS μέτρου [90]:

$$CS_{kernel}(k) = \frac{\sum_{i=1}^{k}\left[\frac{1}{N_i}\sum_{\vec{x}_i \in C_i} max_{\vec{x}_{q \in C_i}}\left\{2\left(1 - K\left(\vec{x}_i, \vec{x}_q\right)\right)\right\}\right]}{\sum_{i=1}^{k}\left[min_{j \in k, j \neq k}\left\{2\left(1 - K\left(\vec{m}_i, \vec{m}_q\right)\right)\right\}\right]}$$

Όπου $K\left(\vec{x}_i - \vec{x}_j\right) = exp\left(-\frac{\vec{x}_i - \vec{x}_j{}^2}{2\sigma^2}\right)$ είναι μια Γκαουσιανή συνάρτηση πυρήνα (kernel function) με $\sigma > 0$. Σκοπός είναι η ελαχιστοποίηση του CS_{kernel} μέτρου. Μικρότερη τιμή του CS_{kernel} υποδηλώνει καλύτερο αποτέλεσμα ομαδοποίησης.

Τα αποτελέσματα του *AKC-BCO* συγκρίθηκαν με αυτά των *Dynamic Clustering using the binary-Particle Swarm Optimization (DCPSO)* [91]*, Dynamic Clustering based on PSO and Genetic algorithm (DCPG)* [92]*, Automatic Kernel Clustering with Multi-Elitist PSO (AKC-MEPSO)* [90] αλγορίθμων, κάνοντας χρήση των εξής συνόλων δεδομένων αναφοράς: Iris, Wine, Glass, Vowel, R15, D31 και bank marketing, με κριτήριο την τιμή του CS_{kernel}. Ως προς τον αριθμό των clusters, ο AKC-BCO είχε αποτέλεσμα πιο κοντά στο πραγματικό από τους άλλους τρεις αλγορίθμους. Από άποψη υπολογιστικού χρόνου, ο AKC-BCO χρειάζεται περισσότερο, και ο λόγος είναι ότι ο AKC-BCO απασχολεί τρεις διαφορετικές μέλισσες, τις “employed”, τις “onlooker” και τις “scout” μέλισσες, σε κάθε επανάληψη, ενώ οι άλλες μέθοδοι έχουν μόνο ένα είδος εργασίας. Σε κάθε επανάληψη, κάθε μέλισσα ενημερώνεται και κάθε στοιχείο δεδομένων αντιστοιχίζεται στον κοντινότερο ενεργό cluster. Ως εκ τούτου, η πολυπλοκότητα του AKC-BCO είναι $O(n^3)$, όπου n είναι ο μεγαλύτερος αριθμός ανάμεσα στον αριθμό επαναλήψεων, τον αριθμό μελισσών και τον αριθμό των στοιχείων δεδομένων (data elements). Επίσης μετρήθηκε το κατά πόσο βελτιώθηκε ο αριθμός των clusters σε σχέση με τους 3 άλλους αλγορίθμους, με χρήση του Λόγου Βελτίωσης (Improvement Ratio):

$$Improvement\ Ratio = \frac{\|\mu_{new} - Opt| - |\mu_{old} - Opt\|}{Opt}$$

Όπου Opt ο γνωστός αριθμός clusters, μ_{old} και μ_{new} ο μέσος αριθμός clusters του εκάστοτε αλγορίθμου και του νέο αλγορίθμου, αντίστοιχα. Υψηλότερο IR φανερώνει μεγαλύτερη βελτίωση. Στις περισσότερες, αλλά όχι όλες, περιπτώσεις, ο AKC-BCO επέφερε σημαντικότερη βελτίωση.

Για την περαιτέρω αξιολόγηση του αλγορίθμου, η προτεινόμενη μέθοδος εφαρμόστηκε και σε ένα πραγματικού κόσμου ιατρικό πρόβλημα, την πρόγνωση καρκίνου του προστάτη. Χρησιμοποιήθηκαν γνωστά ομαδοποιημένα δεδομένα, που αφορούσαν την επιβίωση των ασθενών ύστερα από επέμβαση

στον προστάτη. Ως χαρακτηριστικά ομαδοποίησης ορίστηκαν οι πιο κρίσιμοι παράγοντες: η βιοψία, το ειδικό αντιγόνο του προστάτη (prostate cancer antigen – PSA) και η δακτυλική εξέταση του ορθού (digital rectal examination – DRE). Τα δεδομένα ομαδοποιήθηκαν σε τρεις περιπτώσεις-clusters ανάλογα με το χρόνο επιβίωσης των ασθενών. Ο AKC-BCO προσέγγισε περισσότερο τον αληθινό αριθμό clusters και είχε τη μικρότερη τυπική απόκλιση, γεγονός που συνεπάγεται καλύτερη σταθερότητα. Ο λόγος βελτίωσης ήταν επίσης μεγαλύτερος, όπως και η ακρίβεια, αλλά η τυπική απόκλιση δεν ήταν η μικρότερη. Ωστόσο, ο AKC-BCO συγκλίνει γρηγορότερα και καλύτερα από τον AKC-MEPSO εφόσον έχει μικρότερη τιμή CS_{kernel}. Επιπλέον, αξιολόγηση ως προς την ακρίβεια (Ακρίβεια = Αριθμός σωστά ομαδοποιημένων καταγραφών / Συνολικός αριθμός ομαδοποιημένων καταγραφών) έδειξε πως οι DCPSO και DCPG δεν είναι καθόλου ακριβείς, ενώ ο προτεινόμενος ήταν καλύτερος και από τον AKC-MEPSO. Σε γενικές γραμμές, θα λέγαμε πως ο AKC-BCO είναι ανώτερης ποιότητας των DCPG, DCPSO και AKC-MEPSO.

Κεφάλαιο 5 – Ανασκόπηση

Ανασκόπηση του 4.2

Στο ***"Bias correction fuzzy clustering algorithms"*** [31] βελτιώνονται τα αποτελέσματα ομαδοποίησης αλγορίθμων που λειτουργούν με ασαφή (fuzzy) λογική ενσωματώνοντας σε αυτούς έναν όρο ο οποίος διορθώνει το σφάλμα μεροληψίας (bias correction) που προκύπτει ως συνέπεια της επιλογής μη βέλτιστων αρχικοποιήσεων. Οι προτεινόμενοι Bias-correction FCM (BFCM), Bias-correction Gustafson and Kessel (BGK) και Bias-correction Inter-Cluster Separation (BICS) συγκρίθηκαν με τους Fuzzy C-Means (FCM), Gustafson and Kessel (GK) και Inter-Cluster Separation (ICS). Εξετάστηκαν αρκετές περιπτώσεις και δόθηκαν παραδείγματα τα οποία είχαν ως συμπέρασμα πως σε γενικές γραμμές, αλλά όχι πάντα, οι bias-correction εκδοχές των αλγορίθμων είχαν καλύτερη απόδοση. Ενδεικτικά, το μέσο ποσοστό σφάλματος των FCM, BFCM, GK και BGK ήταν για το σύνολο των banana-shaped δεδομένων 10.78%, 12.34%, 4.38% και 0%, αντίστοιχα. Είναι φανερό πως ο BGK είναι ο πιο αποτελεσματικός στην ανάλυση αυτού του τύπου δεδομένων, όμως, η αποτελεσματικότητα του BFCM δεν είναι υψηλότερη από του απλού FCM.

Ο ***GAKFCM*** [32] είναι συνδυασμός ενός βελτιωμένου γενετικού αλγορίθμου και του Kernel-based fuzzy c-means (KFCM) και στοχεύει και αυτός στη βελτίωση του Fuzzy c-means (FCM). Συγκεκριμένα, επιχειρεί τη βελτίωση του FCM στην ευαισθησία στον θόρυβο, τις αρχικοποιήσεις και την πιθανή σύγκλισή του σε ένα τοπικό ακρότατο. Πράγματι, ο GAKFCM αποδείχτηκε ακριβής και με βελτιωμένη απόδοση.

Ένα άλλο μοντέλο που προτάθηκε για την αντιμετώπιση των αδυναμιών του FCM είναι το ***Credibilistic Clustering Model (CCM)*** [19] το οποίο αξιοποιεί τα θετικά χαρακτηριστικά των μεθόδων FCM και Possibilistic C-Means (PCM) [18], ενώ εξαλείφει τις ατέλειές τους χρησιμοποιώντας το μέτρο αξιοπιστίας [20], το οποίο ορίστηκε στα πλαίσια της θεωρίας αξιοπιστίας [21], έναν κλάδο των μαθηματικών για τη μελέτη των ασαφών φαινομένων.

Στο ***"A new mechanism for RFID clustering and identification"*** [23] παρουσιάζεται ένας μηχανισμός που βασίζεται στον Progressive Scanning (PS) [24] αλγόριθμο και σχεδιάστηκε για την βελτίωση της χαμηλής αποδοτικότητας στην αναγνώριση ετικετών λόγω του προβλήματος της περιπλοκής/σύγκρουσης (collision) των ετικετών στα Radio Frequency Identification (RFID) συστήματα. Ο νέος μηχανισμός συγκρίθηκε με τους *Dynamic Framed Slotted Aloha (DFSA)* [25] και *Enhanced Dynamic Framed Slotted Aloha (EDFSA)* [26] οι οποίοι επιχειρούν το ίδιο. Ο νέος αλγόριθμος αποδείχτηκε ότι μπορεί να επιτύχει καλύτερη επίδοση από τους δύο τελευταίους, μειώνοντας τον απαιτούμενο χρόνο γιγα την αναγνώριση

όλων των ετικετών χάρη στη χρήση διάταξης με δύο αναγνώστες. Επίσης, μειώνει τον αριθμό των συγκρουόμενων ετικετών (collided tags) χάρη στην ομαδοποίηση των ετικετών με τον PS αλγόριθμο.

Ο ***Hybrid density-based clustering for data stream (HDC-Stream)*** [14] αφορά επίσης τα RFID συστήματα και συνδυάζοντας τα πλεονεκτήματα των density grid-based [12] και density-based microclustering [13] μεθόδων κάνει προσπάθεια για τη βελτίωση των αποτελεσμάτων και του χρόνου υπολογισμού. Επίσης, χρησιμοποιείται και ο modified BDSCAN για τον σχηματισμό των τελικών clusters. Ο HDC-Stream παρουσίασε καθαρότητα (purity) ομαδοποίησης 98%, ενώ οι DenStream και D-Stream, οι οποίοι είναι δύο άλλοι γνωστοί αλγόριθμοι, 82% και 78%, αντίστοιχα. Η ανωτερότητα του HDC-Stream επαληθεύεται και με χρήση του μέτρου κανονικοποιημένης αμοιβαίας πληροφορίας (normalized mutual information – NMI). Το ίδιο αποδεικνύεται και στα πειράματα με τα πραγματικά δεδομένα, στα οποία η καθαρότητα του HDC-Stream φτάνει το 95%, ενώ των DenStream και D-Stream 86% και 76%, αντίστοιχα. Ο προτεινόμενος αλγόριθμος έχει επίσης συντομότερο χρόνο εκτέλεσης σε σύγκριση με τους άλλους δύο. Η ελάττωση της χρονικής πολυπλοκότητας επετεύχθη με τη χρήση της grid-based ομαδοποίησης, η οποία μείωσε την χρονική πολυπλοκότητα της συγχώνευσης από $O(mi)$ σε $O(1)$ που είναι ο χρόνος χαρτογράφησης (mapping time). Συνοπτικά, ο HDC-Stream, κατάφερε να βελτιώσει την ποιότητα των αποτελεσμάτων της ομαδοποίησης, καθώς και να μειώσει τον χρόνο υπολογισμού. Παρόλα αυτά δεν κρίνεται κατάλληλος για χρήση σε κατανεμημένα περιβάλλοντα.

Ένας ***νέος clusiVAT*** [15] που αποτελεί επέκταση των ιδεών που παρουσιάστηκαν στο [16] συγκρίθηκε με την απόδοση των των k-means, single pass k-means (spkm), online k-means (okm) και clustering using representatives (CURE). Για compact separated (CS) σύνολα δεδομένων, ο νέος clusiVAT έδωσε 100% ακρίβεια στα αποτελέσματά του σε πολύ λιγότερο χρόνο απ' ό,τι ο k-means και οι παραλλαγές του, καθώς και ο CURE. Για δισδιάστατα μη-CS σύνολα δεδομένων, ο clusiVAT έδωσε υψηλή ακρίβεια (≥99.8%) σε 12-18 φορές συντομότερο χρόνο εκτέλεσης από τον k-means και τις παραλλαγές του και 60-90 φορές συντομότερο από τον CURE. Επίσης, εφαρμόζοντας το Friedman test στα αποτελέσματα PA όπου $PA = \#(Correctly\ labeled\ samples)/\#(Total\ samples)$ και DI (Dunn index) για όλα τα σύνολα δεδομένων, προέκυψε η κατάταξη απόδοσης των αλγορίθμων. Ξεκινώντας από την καλύτερη προς τη χειρότερη απόδοση, η κατάταξη (με τους αντίστοιχους λόγους PA/DI) είναι η εξής: clusiVAT (1.56), spkm (2.17), CURE (2.73), k-means (4.18) και okm (4.36).

Ο ***DBSCAN based on Influence Space and Detecting of border points (ISB-DBSCAN)*** [10] βασίζεται στον DBSCAN, στον οποίο ενσωματώνοντας τρεις νέες μεθόδους, καταφέρνει να βελτιώσει τον χρόνο εκτέλεσης σε βάση δεδομένων μεγάλου μεγέθους (>36.000 σημεία) και να μειώσει τον αριθμό των

απαιτούμενων παραμέτρων. Είναι αναμφισβήτητα καλύτερος όταν χρησιμοποιείται σε σύνολα δεδομένων πολλών διαστάσεων και σημείων, επομένως είναι κατάλληλος για εφαρμογή σε σύνολα δεδομένων με πολυπλοκότερη δομή.

Σκοπός της ***"Clustering Massive Small Data for IoT"*** [39] δημοσίευσης είναι η μείωσης της σπατάλης μνήμης και η βελτίωση της απόδοσης του Hadoop Distributed File System (HDFS) συγχωνεύοντας small data για το σχηματισμό μεγαλύτερων συνόλων δεδομένων τα οποία χειρίζεται αποδοτικότερα το σύστημα.

Ο ***Hybrid k-MICA*** [38] συνδύασε τον k-means και τον *Modified Imperialist Competitive Algorithm (MICA)* με σκοπό να ξεπεράσει το πρόβλημα σύγκλισης σε τοπικά βέλτιστα (local optima) του k-means. Στη σύγκριση του Hybrid k-MICA με διάφορους στοχαστικούς αλγορίθμους, όπως οι MICA, ICA, ACO, PSO, SA, GA, TS, HBMO και k-means κρίθηκε συγκρίσιμος με τους υπόλοιπους αλγορίθμους από την άποψη των καλύτερων, μέσων και χειρότερων λύσεων και της τυπικής απόκλισης. Πέρα από την ισχύ και την αποδοτικότητά του, ο Hybrid k-MICA μπορεί να εφαρμοστεί μόνο όταν ο αριθμός των clusters είναι γνωστός εκ των προτέρων.

Τέλος, προτάθηκαν κάποιοι αλγόριθμοι εμπνευσμένοι από την φύση. Ο ***AntClust*** [33] ομαδοποιεί αισθητήρες με παρόμοια συμφραζόμενα για τη βελτίωση της απόδοσης της context-aware αναζήτησης αισθητήρων στο IoT. Η προτεινόμενη μέθοδος συγκρίθηκε με τον CASSARAM [37] εφόσον είναι ο μόνος αλγόριθμος context-aware αναζήτησης αισθητήρων που λαμβάνει υπόψη ένα μεγάλο εύρος context ιδιοτήτων των αισθητήρων κατά την αναζήτηση. Ο χρόνος εκτέλεσης του AntClust προέκυψε σημαντικά λιγότερος, όμως με ελαφρώς μικρότερη ακρίβεια στην επιλογή αισθητήρων. Η προτεινόμενη μέθοδος αποδείχτηκε επίσης πως παρουσιάζει υψηλή επεκτασιμότητα.

Ο ***Grouping Gravitational Search Algorithm (GGSA)*** [47] ο οποίος βασίζεται στον απλό GSA [46] προσπαθεί να βρει τη βέλτιστη λύση του προβλήματος εμπνευσμένος από τους νόμους του Νεύτωνα για την βαρύτητα και την κίνηση. Η απόδοση του GGSA συγκρίθηκε με την απόδοση των Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Firefly Algorithm (FA) και εννέα ακόμα γνωστές τεχνικές ομαδοποίησης (Bayes Net, Multi Layer Perceptron Artificial Neural Network (MPL-ANN), Radial Basis Function Artificial Neural Network (RBF-ANN), KStar, Bagging, MultiBoostAB, Naïve Bayes Tree (NBTree), Ripple Down Rule (Ridor) και Voting Feature Interval (VFI)). Τα αποτελέσματα επιβεβαίωσαν την αποτελεσματικότητα του GGSA ο οποίος είχε το μικρότερο ποσοστό

εσφαλμένης ταξινόμησης. Μάλιστα, η απόδοσή του σε σχέση με τον PSO και τον GSA ήταν καλύτερη και στα 13 σύνολα δεδομένων, ενώ με τον ABC στα 10 από αυτά.

Ο ***Web Document Clustering based on the Cuckoo Search Algorithm (WDC-CSK)*** [40] βασίζεται στον Cuckoo Search (CS) [41] και ομαδοποιεί τα αποτελέσματα στις μηχανές αναζήτησης για πιο γρήγορη αναζήτηση. Ο WDC-CSK επιτυγχάνει καλύτερο αριθμό clusters σε όλα τα σύνολα δεδομένων του πειράματος και με σημαντική διαφορά. Κατά μέσο όρο, ο WDC-CSK διαφέρει από τον ιδανικό αριθμό clusters κατά 0.93-1.11 ομάδες, ενώ ο Lingo [44] κατά 20.28, ο Suffix Tree Clustering (STC) [45] κατά 6.73 και ο Bisecting k-means κατά 3.74. Επίσης, ο WDC-CSK παρουσιάζει καλύτερα αποτελέσματα από τους άλλους τρεις αλγορίθμους (Lingo, STC, Bisecting k-means) στα εξής μέτρα αξιολόγησης: Precision, Recall, F-measure, Fall-out και Accuracy (Rand index), επομένως αποτελεί βελτίωσή τους. Ο WDC-CSK παρουσιάζει εξαιρετικά αποτελέσματα στα σύνολα δεδομένων αναφοράς και σε σύγκριση με τους διάφορους state of the art αλγορίθμους δείχνει βελτίωση μεταξύ 5.86% και 35.89% στο F-measure, μεταξύ 6.02% και 43.79% στο Recall, μεταξύ 3.67% και 6.82% στο Accuracy και μεταξύ 2.52% και 45.39% στο Fall-out. Τέλος, τα πειράματα έδειξαν βελτίωση και των τιμών του SSL_k μεταξύ 21.70% και 31.76%.

Ανασκόπηση του 4.3

Ένας από τους αλγορίθμους που επιχείρησαν να παρατείνουν τη διάρκεια ζωής των Ασύρματων Δικτύων Αισθητήρων (Wireless Sensor Networks – WSNs) είναι ο αλγόριθμος στο ***"An Energy Balanced Cluster Algorithm for Wireless Sensor Networks"*** [50], ο οποίος 1) εγγυάται ομοιόμορφη κατανομή των cluster-heads και μείωσε την πιθανότητα εμφάνισης του φαινομένου "bottleneck" σε αυτούς μέσω μιας νέας μεθόδου επιλογής των cluster-heads και ενός μονοπατιού inter-cluster επικοινωνίας και 2) αύξησε την αξιοποίηση της διαθέσιμης ενέργειας και παρέτεινε τη διάρκεια ζωής των δικτύων μέσω νέων μεθόδων για την ανακατασκευή των clusters (cluster rebuilding) και την intra-cluster επικοινωνία. Η σύγκρισή του έγινε με δύο αλγορίθμους. Ο πρώτος είναι ο *Deterministic Cluster-Head Selection (DCHS)* [49], ο οποίος είναι ένας βελτιωμένος αλγόριθμος με βάση τον LEACH [48] που χρησιμοποιεί την παράμετρο της ενέργειας για να επηρεάσει την επιλογή cluster-head έτσι ώστε να εγγυηθεί ότι οι κόμβοι έχουν αρκετή ενέργεια για να λειτουργήσουν ως cluster-heads. Ο δεύτερος είναι ο "Opt Algorithm" [51] στον οποίο οι cluster-heads εναλλάσσονται εκ περιτροπής ώστε να κατανεμηθεί ομοιόμορφα τον ενεργειακό φόρτο (energy load) σε όλους τους κόμβους. Με τον DCHS και τον Opt η ενέργεια του δικτύου εξαντλήθηκε ύστερα από 586 και 881 γύρους αντίστοιχα, ενώ με τον βελτιωμένο αλγόριθμο ύστερα από 1089, οπότε ο τελευταίος χρησιμοποιεί πιο αποτελεσματικά την ενέργεια του δικτύου. Ύστερα συγκρίθηκαν ως προς

την αναλογία ανακατασκευή των clusters/διάρκεια ζωής του δικτύου. Ο Opt και ο DCHS είχαν ίση διάρκεια ζωής με τις φορές που έγινε ανακατασκευή των clusters, ενώ ο προτεινόμενος αλγόριθμος μειώνει σημαντικά τις φορές που πραγματοποιείται cluster rebuilding. Τέλος, παρατηρήθηκε ότι ο προτεινόμενος αλγόριθμος καθυστερεί σημαντικά τον «θάνατο» των κόμβων εξισορροπώντας την κατανάλωση ενέργειας και παρατείνει την διάρκεια ζωής του δικτύου.

Η ***Node Density based Clustering and Mobile Collection (NDCMC)*** [59] είναι μια υβριδική μέθοδος η οποία με σκοπό την παράταση της διάρκειας ζωής των WSNs συνδύασε την ιεραρχική δρομολόγηση με την συλλογή δεδομένων με ένα κινητό στοιχείο (mobile element – ME). Ένας αριθμός cluster-heads συγκεντρώνει πληροφορίες από τα μέλη των clusters και ύστερα ένα ME επισκέπτεται τους cluster-heads για να συλλέξει τα δεδομένα. Η επιλογή των cluster-heads γίνεται με βάση την πυκνότητα και αυτή η στρατηγική αύξησε τη διάρκεια ζωής του δικτύου κατά 50% σε σχέση με το *Random Clustering and Mobile Collection (RCMC)* σχέδιο που παρουσιάστηκε στο ίδιο άρθρο και το οποίο τους επιλέγει τυχαία. Σε σύγκριση με άλλες τεχνικές που χρησιμοποιούν MEs – τις MILP, CSPLI και SST [60] – ο NDCMC είχε σημαντικά καλύτερη απόδοση στην παράταση ζωής του δικτύου. Επίσης, αποδείχτηκε ότι είναι πιο οικονομικός ενεργειακά και με πιο εξισορροπημένη κατανάλωση ενέργειας σε σχέση με κινητά στοιχεία σταθερής τροχιάς (fixed ME tracks). Τέλος, σε σύγκριση με αμιγώς βασισμένη-στην-ομαδοποίηση συλλογή δεδομένων, και πιο συγκεκριμένα σε σύγκριση με τον LEACH [55] στον οποίο οι CHs επιλέγονται τυχαία και περιοδικά, ο NDCMC σημείωσε περισσότερο χρόνο λειτουργίας για τον ίδιο αριθμό κόμβων με εναπομείνασα ενέργεια πάνω από ένα κατώφλι.

Ένας άλλος αλγόριθμος που συγκρίθηκε με τον LEACH, αλλά και τον *Energy Aware Multi-Hop Multi-Path Hierarchy (EAMMH)* είναι ο *Improved M2M Clustering Process (IMPC)* [57]. Αν και είναι κάπως σύνθετος, έχει καλύτερη απόδοση όσον αφορά την κατανάλωση ενέργειας, τη διάρκεια ζωής του δικτύου και τον αριθμό κόμβων που εξαντλούν την ενέργειά τους.

Ο ***Service-Aware Clustering (SAC)*** [61] έλαβε υπόψη του τις διαφορετικές υπηρεσίες που προσφέρουν οι κόμβοι-αισθητήρες και βελτίωσε την ενεργειακή ανισορροπία που προκύπτει γι’ αυτό το λόγο. Η διάρκεια ζωής του δικτύου αξιολογήθηκε με βάση τον πρώτο και τον τελευταίο κόμβο που εξαντλεί την ενέργειά του – first node to die (FND) και last node to die (LND), αντίστοιχα. Ο σταθμός βάσης τοποθετήθηκε πρώτα στο κέντρο του δικτύου. Οι FND προέκυψαν στον 332 γύρο για τον DECSA, στον 391 για τον MOCRN και στον 448 για τον SAC. Οι LND διατήρησαν την ενέργειά τους μέχρι τους γύρους 728, 831 και 968 για τους DECSA, MOCRN και SAC, αντίστοιχα. Η διαφορά γίνεται μεγαλύτερη όταν ο σταθμός βάσης τοποθετήθηκε έξω από την περιοχή του δικτύου. Τότε οι FND για τους DECSA, MOCRN

και SAC προέκυψαν στους γύρους 255, 298 και 361, ενώ οι LND στους 556, 746 και 802, αντίστοιχα. Ο SAC, λοιπόν, τρέχει για περισσότερους γύρους προτού ο πρώτος και ο τελευταίος κόμβος εξαντλήσουν την ενέργειά τους. Ο λόγος είναι ότι έλαβε υπόψη παραμέτρους όπως την τρέχουσα ενέργεια των κόμβων-αισθητήρων, τη διαφοροποίηση των υπηρεσιών και την απόσταση μεταξύ κόμβων κατά την επιλογή νέων cluster-heads. Η βελτίωση στη διάρκεια ζωής του δικτύου μετρήθηκε και με βάση τη μέση διάρκεια ζωής. Η μέση διάρκεια ζωής που πέτυχε ο SAC για 100 κόμβους είναι 10.8% υψηλότερη από του DECSA και 2.4% από του MOCRN. Σε πείραμα με 500 κόμβους η μέση διάρκεια ζωής του SAC προέκυψε 23.7% υψηλότερη από του DECSA και 5.3% από του MOCRN. Επίσης, ο SAC σχηματίζει λιγότερους clusters και αυτό συνεπάγεται ότι είναι περισσότερο επεκτάσιμος.

Ο ***Density-based Energy-efficient Clustering Algorithm (DECA)*** [64] όπως και ο NDCMC, αλλά και ο SAC, εγγυάται ομοιόμορφη κατανομή των cluster-heads καθώς για την επιλογή τους, εκτός από την εναπομείνασα ενέργεια των κόμβων, λαμβάνει υπόψη του και την πυκνότητά τους. Η τακτική αυτή σε συνδυασμό με έναν νέο αλγόριθμο που αναπτύχθηκε για multi-hop inter-routing μείωσε την κατανάλωση ενέργειας και επέκτεινε τη διάρκεια ζωής του WSN. Ο DECA συγκρίθηκε με τον LEACH και τον Density-based Clustering Protocol (DBCP) που αποτελεί βελτίωση του LEACH και επιβεβαιώθηκε η καλύτερη ενεργειακή απόδοσή του. Μάλιστα, στον LEACH ο πρώτος κόμβος εξάντλησε την ενέργειά του στον 94 γύρο, ο DBCP στον 124 και ο DECA στον 248 – που είναι διπλάσιος αριθμός γύρων από την περίπτωση του DBCP. Επομένως, ο DECA βελτίωσε σημαντικά την διάρκεια ζωής του δικτύου.

Τέλος, για τον ίδιο σκοπό αναπτύχθηκε μια διαφορετική μέθοδος, ένα ***Energy-Efficient Hierarchical Clustering index tree (ECH-tree)*** [52] βασισμένο στο “grid cell clustering”. Η περιοχή του WSN χωρίζεται ομοιόμορφα σε κελιά και στη συνέχεια ομαδοποιούνται εξασφαλίζοντας ότι η ενέργεια για την προώθηση μηνυμάτων μεταξύ τους θα είναι η ελάχιστη. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να σχηματιστεί ένα ιεραρχικό δέντρο. Αξιοποιώντας αυτό το ECH-tree μετριάζεται η κατανάλωση ενέργειας. Σε γενικές γραμμές, παρατηρήθηκε πως η νέα μέθοδος κατανάλωνε λιγότερη ενέργεια από την παραδοσιακή μέθοδο και πως η απόδοση ήταν καλύτερη ειδικά όταν οι αισθητήρες ήταν περισσότεροι και πιο πυκνά τοποθετημένοι.

Ανασκόπηση του 4.4

Ο ***Fast Modified Global k-means (FMGKM)*** [65] αλγόριθμος αναπτύχθηκε με σκοπό τη μείωση της χρήσης της μνήμης (memory usage) των global k-means (GKM) και modified global k-means (MGKM) αλγορίθμων

που αποτελούν παραλλαγές του k-means. Ο FMGKM χρειάζεται λιγότερη υπολογιστική προσπάθεια από τον GKM και τον MGKM. Με τον αλγόριθμο Fast global k-means clustering using cluster membership and inequality [66] έχουν παρόμοια υπολογιστική πολυπλοκότητα. Ο προτεινόμενος FMGKM αποδείχτηκε στις περισσότερες περιπτώσεις γρηγορότερος και ακριβέστερος από τον GKM. Τα αποτελέσματά του ήταν παρόμοια με του MGKM, προέκυψαν όμως σε λιγότερο χρόνο επεξεργασίας (CPU time). Η βελτίωση που προσφέρει ο *Fast Modified Global k-means (FMGKM)* γίνεται όλο και πιο ουσιαστική όταν το μέγεθος του συνόλου δεδομένων αυξάνει.

Ο αλγόριθμος ***Dynamic k-nearest-neighbor (DKNNA)*** [67] επιτυγχάνει τα ίδια σχεδόν αποτελέσματα ομαδοποίησης με τον *Fast Pairwise Nearest Neighbor (FPNN)* [70]. Συγκρινόμενος με τον FPNN με γρήγορη αναζήτηση (Fast Search – FS) για την εύρεση των κοντινότερων γειτόνων (FPNN + FS), η προτεινόμενη μέθοδος σε συνδυασμό με τον ίδιο αλγόριθμο γρήγορης αναζήτησης (DKNNA + FS) μείωσε τον χρόνο υπολογισμού κατά 1.90-2.18 για το σύνολο δεδομένων από μια πραγματική εικόνα και κατά 1.92-2.02 χρησιμοποιώντας το σύνολο δεδομένων που παράγεται από τρεις εικόνες. Επίσης, ο DKNNA + FS μπορεί να ελαττώσει το μέσο τετραγωνικό σφάλμα κατά 1.26% για το ίδιο σύνολο δεδομένων.

Ο ***Hyperellipsoidal clustering for resource-constrained environments (HyCARCE)*** [71] έχει πολυπλοκότητα $O(N)$ και αναπτύχθηκε για περιβάλλοντα με υπολογιστικούς περιορισμούς. Για τη σύγκριση χρησιμοποιήθηκε ο *Subtractive Clustering (SC)* [73] διότι είναι παρόμοιος με τον HyCARCE ως προς τον τρόπο με τον οποίο βρίσκει το κέντρο των clusters και το όριο απόφασης (decision boundary) για την ομαδοποίηση των δεδομένων και ο *Gustafson-Kessel (GK)* [72] επειδή είναι ένας προηγμένος ασαφής (fuzzy) αλγόριθμος που χρησιμοποιήθηκε ως σημείο αναφοράς για την ακρίβεια. Ο HyCARCE συγκρίθηκε και με τους πιο κλασικούς DENCLUE και k-means. Η υπολογιστική πολυπλοκότητα του HyCARCE προέκυψε συγκρίσιμη με αυτή των k-means, DENCLUE και GK, ενώ ο SC είναι υπολογιστικά ακριβότερος. Επίσης, η ακρίβεια του HyCARCE προέκυψε συγκρίσιμη ή καλύτερη από εκείνη του GK. Ως προς το χρόνο εκτέλεσης, ο HyCARCE αποδείχτηκε σημαντικά ταχύτερος από τον DENCLUE και τον GK, για όλα τα σύνολα δεδομένων του πειράματος – χρειάστηκε κατά μέσο όρο χρόνο εκτέλεσης (CPU time) 0.28 δευτερόλεπτα, ενώ ο DENCLUE 11 δευτερόλεπτα. Τέλος, ο HyCARCE αποδείχτηκε λιγότερο ευαίσθητος στην επιλογή παραμέτρων από τον DENCLUE, όμως η ισχύς των αποτελεσμάτων επηρεάζεται αρνητικά σε υψηλότερες διαστάσεις, γι' αυτό και κρίθηκε καταλληλότερος για δεδομένα χαμηλών διαστάσεων.

Τέλος, ένας νέος αλγόριθμος για την ομαδοποίηση συνεχών δεδομένων ή δεδομένων διαστήματος (interval data), ***"A new topological clustering algorithm for interval data"*** [74], κατάφερε να αποδειχτεί

ποιοτικά καλύτερος από τους DIV και SCLUST που είναι αλγόριθμοι για τον ίδιο σκοπό και παρόμοιος με τη μέθοδο SHICLUST. Η πολυπλοκότητά του είναι γραμμική – κάτι που δεν ισχύει για τον SHICLUST – επομένως προσφέρει άριστα αποτελέσματα σε μικρό χρόνο επεξεργασίας. Σύγκριση έγινε και με τις μεθόδους SYKSOM και SYKCLUST. Ο SYKSOM δεν καταφέρνει να τον ανταγωνιστεί, ενώ τα αποτελέσματα είναι παρόμοια με του SYKCLUST, στον οποίο όμως απαιτείται ο προσδιορισμός του αριθμού των clusters εκ των προτέρων.

Ο ***Pattern Reduction (PR)*** [77] λειτουργεί σε συνδυασμό με τον k-means και άλλους βασισμένους σε αυτόν αλγορίθμους, με σκοπό τη μείωση του χρόνου υπολογισμού. Πιο συγκεκριμένα, εφαρμόστηκε στους εξής 5 κλασικούς αλγορίθμους ομαδοποίησης: k-means (KM), Relational k-means (RKM), Kernel k-means (KKM), Triangle inequality k-means (TKM) και Genetic k-means (GKA) ώστε να αξιολογηθεί η απόδοσή του. Ο PR μείωσε τον χρόνο υπολογισμού των KM, RKM, TKM, KKM και GKA κατά 79%, 51%, 81%, 82% και 74%, αντίστοιχα, αν και με μια μικρή απώλεια στην ποιότητα.

Ο ***Optimal FJP (OFJP)*** [80] κατάφερε να βελτιώσει την ταχύτητα των επιτυχημένων μεθόδων Fuzzy Joint Points (FJP) [78] και Modified FJP (MFJP), ενσωματώνοντας σε αυτές κάποιες μεθόδους που αντικαθιστούν τα χρονοβόρα βήματά τους. Ο OFJP τρέχει σε $O(n^2)$ χρόνο και τα αποτελέσματα των προσομοιώσεων έδειξαν πως η προτεινόμενη μέθοδος είναι δραματικά γρηγορότερη από τον MFJP. Στην ίδια δημοσίευση προτείνεται και ένας ακόμα αλγόριθμος, ο αScan [80], ο οποίος είναι ακόμα πιο γρήγορος, εξαρτάται όμως από μία παράμετρο εισόδου.

Ανασκόπηση του 4.5

Από τους αλγορίθμους που αναπτύχθηκαν για τη βελτίωση της ποιότητας της ομαδοποίησης, ο ***Induction and Deduction Clustering (IDC)*** [87] αλγόριθμος κατάφερε μέσω του συνδυασμού των μεθόδων επαγωγής και παραγωγής να μειώσει την αβεβαιότητα που προκύπτει από την επαγωγή που χρησιμοποιούν οι αλγόριθμοι ομαδοποίησης, εφόσον ομαδοποιούν αντικείμενα δεδομένων σε clusters σύμφωνα με τις ομοιότητές τους. Στα πειράματα συγκρίθηκε με τον BIRCH και τον K-means και πράγματι, αποδείχτηκε πως έχει υψηλότερη ακρίβεια ομαδοποίησης και ικανότητα ανίχνευσης outliers. Ωστόσο, απαιτεί από τον χρήστη τον προσδιορισμό κάποιων συντελεστών και χρειάζεται να ενισχυθεί η ικανότητά του αντιμετώπισης δεδομένων υψηλών διαστάσεων.

Ο ***Hierarchical RFID Trajectory Clustering (HRTC)*** αλγόριθμος ο οποίος αναπτύχθηκε για να χειριστεί τις αβεβαιότητες σε εφαρμογές ιχνηλασιμότητας με RFID, συγκρίθηκε με τους Time-Focused Clustering (TFC)

[86] και Fuzzy C-Means (FCM) αλγορίθμους. Ο HRTC είχε υψηλότερες επιδόσεις ως προς την ποιότητα ομαδοποίησης όταν ο αριθμός των clusters ήταν αρκετά μεγάλος – 70 εν προκειμένω. Στην αποδοτικότητα από άποψη χρόνου ο TFC ήταν καλύτερος, αυτό συνέβη όμως διότι δε δίνει βάση στις αβεβαιότητες των δεδομένων οπότε δεν σπαταλά επιπλέον χρόνο σε αυτές. Από τους άλλους δύο αλγορίθμους οι οποίοι υπολογίζουν τις αβεβαιότητες, ο HRTC αποδείχτηκε καλύτερος. Σε μη υψηλό επίπεδο αβεβαιότητας, ο HRTC και ο TFC είχαν παρόμοιο χρόνο τρεξίματος.

Ο ***Automatic Kernel Clustering with Bee Colony Optimization (AKC-BCO)*** [88] συγκρίθηκε με τους *Dynamic Clustering using the binary-Particle Swarm Optimization (DCPSO)* [91]*, Dynamic Clustering based on PSO and Genetic algorithm (DCPG)* [92]*, Automatic Kernel Clustering with Multi-Elitist PSO (AKC-MEPSO)* [90] αλγορίθμους και τα αποτελέσματά του ως προν τον αριθμό των clusters ήταν γενικά πιο κοντά στον πραγματικό αριθμό των clusters των πειραματικών δεδομένων αναφοράς. Πιο συγκεκριμένα, η βελτίωση που επέφερε ήταν από 9.33% έως 75.00% σε σχέση με τον AKC-MEPSO, από 0.00% έως 48.83% σε σχέση με τον DCPG και τέλος από 0.00% έως 40.50% σε σχέση με τον DCPSO. Παρόλα αυτά, ο AKC-BCO χρειάζεται περισσότερο υπολογιστικό χρόνο από τις άλλες μεθόδους. Η μεγαλύτερη ακρίβεια του AKC-BCO ως προς τον προσδιορισμό του αριθμού των clusters αποδείχτηκε και με τη χρήση δεδομένων από ένα πρόβλημα πραγματικού κόσμου σχετικό με τον καρκίνο του προστάτη. Επίσης, η σύγκλισή του ήταν γρηγορότερη και καλύτερη από του AKC-MEPSO.

Κεφάλαιο 6 – Επίλογος

Με μια γρήγορη εξέταση των αποτελεσμάτων στις μηχανές αναζήτησης επιστημονικών άρθρων γίνεται φανερό ότι οι τεχνικές ομαδοποίησης (clustering techniques) για την εξόρυξη δεδομένων γενικά, αλλά και ειδικότερα από το Διαδίκτυο των Πραγμάτων, είναι ένα πεδίο που έχει προσελκύσει το ενδιαφέρον των ερευνητών. Οι περισσότεροι παραδοσιακοί αλγόριθμοι εξόρυξης δεδομένων δεν μπορούν να εφαρμοστούν απευθείας για την επεξεργασία του μεγάλου όγκου δεδομένων του Internet of Things. Έτσι, οι νέες μέθοδοι ομαδοποίησης στοχεύουν στην βελτίωση της απόδοσης, στη μείωση της πολυπλοκότητας και στη γενικότερη βελτίωση της ποιότητας των αλγορίθμων, καθώς και στην μείωση της κατανάλωσης ενέργειας στα WSNs, με απώτερο σκοπό την αποτελεσματική και αποδοτική υλοποίηση του Internet of Things.

Με βάση την αναζήτηση επιστημονικών άρθρων που πραγματοποιήθηκε στα πλαίσια της παρούσας εργασίας, διαπιστώθηκε πως οι προτεινόμενες τεχνικές είναι κυρίως προεκτάσεις άλλων υπαρχόντων μεθόδων ή/και συνδυασμός τους και πιο σπάνια νέες ιδέες. Οι περισσότεροι αλγόριθμοι πετυχαίνουν σημαντικές βελτιώσεις, αν και πολλές φορές υπάρχουν «παράπλευρες απώλειες» και πρέπει να γίνει κάποιου είδους συμβιβασμός (tradeoff) ανάμεσα στα χαρακτηριστικά, ανάλογα με το ζητούμενο και τις προτεραιότητες της εκάστοτε εφαρμογής. Παραδείγματος χάριν, ο AntClust [33] βελτίωσε σημαντικά την ταχύτητα της context-aware αναζήτησης αισθητήρων, αλλά με μια απώλεια στην ακρίβεια του αποτελέσματος. Ένα άλλο παράδειγμα είναι ο AKC-BCO [88], ο οποίος κατάφερε με πολύ ακριβή αποτελέσματα να αυτοματοποιήσει την διαδικασία ομαδοποίησης (να μην απαιτείται, δηλαδή, ο ορισμός του αριθμού των ομάδων εκ των προτέρων), χρειάζεται όμως περισσότερο χρόνο υπολογισμού. Πάντως, μείζονος σημασίας είναι η ανάπτυξη αλγορίθμων που κατά κύριο λόγο μπορούν να χειριστούν μεγάλα σύνολα δεδομένων και μάλιστα υψηλών διαστάσεων. Ειδάλλως, η υλοποίηση του Internet of Things όπως αυτό ορίζεται, δεν είναι εφικτή.

Βιβλιογραφία

[1] P. Andritsos, “Data clustering techniques,” *Toronto, Univ. Toronto, Dep. Comput. …*, 2002.

[2] A. K. Jain and R. C. Dubes, “Algorithms for Clustering Data,” *Prentice Hall*, vol. 355. p. 320, 1988.

[3] a. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[4] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. 2012.

[5] L. Kaufman and P. J. Rousseeuw, “Finding Groups in Data: An Introduction to Cluster Analysis,” *Wiley-Interscience*, vol. 33, no. 1, p. 368, 2005.

[6] J. H. Raymond T Ng, “Efficient and effective clustering methods for spatial data mining,” *Proc. Int. Conf. Very Large Data Bases*, pp. 144–155, 1994.

[7] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An Efficient Data Clustering Databases Method for Very Large,” *ACM SIGMOD Int. Conf. Manag. Data*, vol. 1, pp. 103–114, 1996.

[8] S. Guha, R. Rastogi, and K. Shim, “Cure,” *Proc. 1998 ACM SIGMOD Int. Conf. Manag. data - SIGMOD ’98*, pp. 73–84, 1998.

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Proc. Second Int. Conf. Knowl. Discov. Data Min. (KDD-96), Portland, Oregon, USA*, pp. 226–231, 1996.

[10] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, “An efficient and scalable density-based clustering algorithm for datasets with complex structures,” *Neurocomputing*, vol. 171, pp. 9–22, 2016.

[11] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander, “OPTICS : Ordering Points To Identify the Clustering Structure,” *SIGMOD ’99 Proc. 1999 ACM SIGMOD Int. Conf. Manag. data*, vol. 28, no. 2, pp. 49–60, 1999.

[12] A. Amini, T. Y. Wah, M. R. Saybani, and S. R. A. S. Yazdi, “A study of density-grid based clustering algorithms on data streams,” *2011 Eighth Int. Conf. Fuzzy Syst. Knowl. Discov.*, vol. 3, pp. 1652–1656, 2011.

[13] A. Amini and T. Wah, “Density Micro-Clustering Algorithms on Data Streams: A Review,” *Proc. Int. MultiConference …*, vol. I, pp. 14–18, 2011.

[14] A. Amini, H. Saboohi, T. Ying Wah, and T. Herawan, “A Fast Density-Based Clustering Algorithm for Real-Time Internet of Things Stream,” *Sci. World J.*, vol. 2014, pp. 1–11, 2014.

[15] D. Kumar, J. C. Bezdek, L. Fellow, M. Palaniswami, S. Rajasegarar, C. Leckie, and T. C. Havens, “A Hybrid Approach to Clustering in Big Data,” pp. 1–14, 2015.

[16] T. C. Havens, J. C. Bezdek, and M. Palaniswami, “Scalable single linkage hierarchical clustering for big data,” *Proc. 2013 IEEE 8th Int. Conf. Intell. Sensors, Sens. Networks Inf. Process. Sens. Futur. ISSNIP 2013*, vol. 1, pp. 396–401, 2013.

[17] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Comput.*

Geosci., vol. 10, no. 2–3, pp. 191–203, 1984.

[18] R. Krishnapuram and J. M. Keller, "A Possibilistic Approach to Clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, 1993.

[19] M. R. N. Kalhori, M. H. F. Zarandi, and I. B. Turksen, "A new credibilistic clustering algorithm," *Inf. Sci. (Ny).*, vol. 279, pp. 105–122, 2014.

[20] B. Liu and Y. K. Liu, "Expected value of fuzzy variable and fuzzy expected value models," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 4, pp. 445–450, 2002.

[21] B. Liu, *Uncertainty Theory: An Introduction to Its Axiomatic Foundations*. Springer-Verlag Berlin, 2004.

[22] J. Zhou, X. Wang, C.-C. Hung, and S. Chen, "Fuzzy clustering based on credibility measure," *Proc. Sixth Int. Conf. Inf. Manag. Sci. , Lhasa, China*, pp. 404–411, 2007.

[23] O. Fathia, "A new mechanism for RFID clustering and identification," 2014.

[24] W. Su, N. Alchazidis, and T. T. Ha, "Multiple RFID Tags Access Algorithm," *IEEE Trans. Mob. Comput.*, vol. 9, no. 2, pp. 174–187, 2010.

[25] F. C. Schoute, "Dynamic Frame Length ALOHA," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565–568, 1983.

[26] S. R. Lee, S. D. Joo, and C. W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," *MobiQuitous 2005 Second Annu. Int. Conf. Mob. Ubiquitous Syst. - Networking Serv.*, pp. 166–172, 2005.

[27] W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3918 LNAI, pp. 577–593, 2006.

[28] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: Parameter reduction and outlier detection," *Inf. Syst.*, vol. 38, no. 3, pp. 317–330, 2013.

[29] T. N. Tran, T. T. Nguyen, T. A. Willemsz, G. van Kessel, H. W. Frijlink, and K. van der V. Maarschalk, "A density-based segmentation for 3D images, an application for X-ray micro-tomography," *Anal. Chim. Acta*, vol. 725, pp. 14–21, 2012.

[30] J. Yu and M. S. Yang, "Optimality test for generalized FCM and its application to parameter selection," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 1, pp. 164–176, 2005.

[31] M.-S. Yang and Y.-C. Tian, "Bias-correction fuzzy clustering algorithms," *Inf. Sci. (Ny).*, vol. 309, pp. 138–162, 2015.

[32] Y. Ding and X. Fu, "Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm," *Neurocomputing*, pp. 1–6, 2015.

[33] M. Ebrahimi, E. ShafieiBavani, R. K. Wong, S. Fong, and J. Fiaidhi, "An adaptive meta-heuristic search for the internet of things," *Futur. Gener. Comput. Syst.*, 2015.

[34] E. D. Lumer and B. Faieta, "Diversity and adaptation in populations of clustering ants," *Proc. Third Int. Conf. Simul. Adapt. Behav. From Anim. to Animat. 3, MIT Press. Cambridge, MA, USA*, pp. 501–508, 1994.

[35] J. Handl and B. Meyer, “Improved ant-based clustering and sorting in a document retrieval interface,” *Parallel Probl. Solving from Nature—PPSN VII*, pp. 913–923, 2002.

[36] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, “The Dynamics of Collective Sorting Robot . Like Ants and Ant . Like Robots,” *Proc. first Int. Conf. Simul. Adapt. Behav. From Anim. to Animat.*, pp. 356–363, 1991.

[37] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Christen, and D. Georgakopoulos, “Sensor search techniques for sensing as a service architecture for the internet of things,” *IEEE Sens. J.*, vol. 14, no. 2, pp. 406–420, 2014.

[38] T. Niknam, E. Taherian Fard, N. Pourjafarian, and A. Rousta, “An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering,” *Eng. Appl. Artif. Intell.*, vol. 24, no. 2, pp. 306–317, 2011.

[39] X. Tao and C. Ji, “Clustering massive small data for IOT,” *2014 2nd Int. Conf. Syst. Informatics (ICSAI 2014)*, no. Icsai, pp. 974–978, 2014.

[40] C. Cobos, H. Muñoz-Collazos, R. Urbano-Muñoz, M. Mendoza, E. León, and E. Herrera-Viedma, “Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion,” *Inf. Sci. (Ny).*, vol. 281, pp. 248–264, 2014.

[41] X. S. Yang and S. Deb, “Cuckoo search via Lévy flights,” *2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc.*, pp. 210–214, 2009.

[42] X. Yang, *Nature-Inspired Metaheuristic Algorithms*. 2010.

[43] N. Bacanin, “An object-oriented software implementation of a novel cuckoo search algorithm,” *Proc. 5th Eur. Conf. Eur. Comput. Conf. , World Sci. Eng. Acad. Soc.*, pp. 245–250, 2011.

[44] S. Osiński and D. Weiss, “A Concept-Driven Algorithm for Clustering Search Results,” *IEEE Intell. Syst.*, 2005.

[45] O. Zamir and O. Etzioni, “Web Document Clustering: A Feasibility Demonstration,” *Sigir*, pp. 46–54, 1998.

[46] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” *Inf. Sci. (Ny).*, vol. 179, no. 13, pp. 2232–2248, 2009.

[47] M. B. Dowlatshahi and H. Nezamabadi-pour, “GGSA: A Grouping Gravitational Search Algorithm for data clustering,” *Eng. Appl. Artif. Intell.*, vol. 36, pp. 114–121, 2014.

[48] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 00, no. c, pp. 3005–3014, 2000.

[49] M. J. Handy, M. Haase, and D. Timmermann, “Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection,” *4th Int. Work. Mob. Wirel. Commun. Netw.*, pp. 368–372, 2002.

[50] Z. Zhou and T. Wang, “An Energy Balanced Clustering Algorithm for Wireless Sensor Networks,” 2012.

[51] M. C. M. Thein and T. Thein, “An Energy Efficient Cluster-Head Selection for Wireless Sensor

Networks," *2010 Int. Conf. Intell. Syst. Model. Simul.*, pp. 287–291, 2010.

[52] J. Tang, Z. Zhou, J. Niu, and Q. Wang, "An energy efficient hierarchical clustering index tree for facilitating time-correlated region queries in the Internet of Things," *J. Netw. Comput. Appl.*, vol. 40, pp. 1–11, 2014.

[53] I. Peng and Y. Chen, "Energy consumption bounds analysis and its applications for grid based wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 444–451, 2013.

[54] S.-W. Han, I.-S. Jeong, and S.-H. Kang, "Low latency and energy efficient routing tree for wireless sensor networks with multiple mobile sinks," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 156–166, 2013.

[55] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, pp. 660–670, 2002.

[56] H. O. Tan, I. Korpeoglu, and I. Stojmenovic, "Computing Localized Power-Efficient Data Aggregation Trees for Sensor Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 3, pp. 489–500, 2011.

[57] K. V. Deshpande, "Design of an Improved Energy Efficient Clustering in M2M Communication," 2015.

[58] P. Zhang and G. Miao, "Energy-Efficient Clustering Design for M2M Communications," *2014 IEEE Glob. Conf. Signal Inf. Process. Glob. 2014*, no. 1, pp. 163–167, 2014.

[59] R. Zhang, J. Pan, J. Liu, D. Xie, and B. Columbia, "NDCMC: A Hybrid Data Collection Approach for Large-Scale WSNs Using Mobile Element and Hierarchical Clustering," pp. 1584–1589, 2015.

[60] W. Liang, J. Luo, and X. Xu, "Prolonging Network Lifetime via A Controlled Mobile Sink in Wireless Sensor Networks," *GLOBECOM - IEEE Glob. Telecommun. Conf.*, 2010.

[61] A. Bagula, A. Abidoye, and G.-A. Zodi, "Service-Aware Clustering: An Energy-Efficient Model for the Internet-of-Things," *Sensors*, vol. 16, no. 1, p. 9, 2015.

[62] Z. Yong and Q. Pei, "A energy-efficient clustering routing algorithm based on distance and residual energy for Wireless Sensor Networks," *Procedia Eng.*, vol. 29, pp. 1882–1888, 2012.

[63] C. S. Nam, S. T. Bae, J. W. Chung, and D. R. Shin, "Multihop-based optimal cluster heads numbers considering relay node in transmission range of sensor nodes in wireless sensor networks," *Int. J. Distrib. Sens. Networks*, vol. 2013, 2013.

[64] Z. Xu, Y. Yin, and J. Wang, "A Density-based Energy-efficient Clustering Algorithm for Wireless Sensor Networks," *Int. J. Futur. Gener. Commun. Netw.*, vol. 6, no. 1, pp. 75–86, 2013.

[65] A. M. Bagirov, J. Ugon, and D. Webb, "Fast modified global k-means algorithm for incremental cluster construction," *Pattern Recognit.*, vol. 44, no. 4, pp. 866–876, 2011.

[66] J. Z. C. Lai and T. J. Huang, "Fast global k-means clustering using cluster membership and inequality," *Pattern Recognit.*, vol. 43, no. 5, pp. 1954–1963, 2010.

[67] J. Z. C. Lai and T.-J. Huang, "An agglomerative clustering algorithm using a dynamic k-nearest-neighbor list," *Inf. Sci. (Ny).*, vol. 181, no. 9, pp. 1722–1734, 2011.

[68] J. H. Ward, “Hierarchical grouping to optimize an objective function,” *Journal of the American Statistical Association*, vol. 58, no. 301. pp. 236–244, 1963.

[69] P. Fränti, O. Virmajoki, and V. Hautamäki, “Fast agglomerative clustering using a k-nearest neighbor graph,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1875–1881, 2006.

[70] P. Fränti, T. Kaukoranta, D. F. Shen, and K. S. Chang, “Fast and memory efficient implementation of the exact PNN,” *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 773–777, 2000.

[71] M. Moshtaghi, S. Rajasegarar, C. Leckie, and S. Karunasekera, “An efficient hyperellipsoidal clustering algorithm for resource-constrained environments,” *Pattern Recognit.*, vol. 44, no. 9, pp. 2197–2209, 2011.

[72] D. Gustafson and W. Kessel, “Fuzzy clustering with a fuzzy covariance matrix,” *1978 IEEE Conf. Decis. Control Incl. 17th Symp. Adapt. Process.*, no. 2, pp. 761–766, 1978.

[73] S. L. Chiu, “Fuzzy Model Identification Based on Cluster Estimation,” *Journal of intelligent and Fuzzy systems*, vol. 2, no. 3. pp. 267–278, 1994.

[74] G. Cabanes, Y. Bennani, R. Destenay, and A. Hardy, “A new topological clustering algorithm for interval data,” *Pattern Recognit.*, vol. 46, no. 11, pp. 3030–3039, 2013.

[75] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag Berlin, 2001.

[76] G. Cabanes and Y. Bennani, “A Simultaneous Two-Level Clustering Algorithm for Automatic Model Selection,” *Proc. - 6th Int. Conf. Mach. Learn. Appl. ICMLA 2007*, pp. 375–380, 2007.

[77] M. C. Chiang, C. W. Tsai, and C. S. Yang, “A time-efficient pattern reduction algorithm for k-means clustering,” *Inf. Sci. (Ny).*, vol. 181, no. 4, pp. 716–731, 2011.

[78] E. N. Nasibov and G. Ulutagay, “A new unsupervised approach for fuzzy clustering,” *Fuzzy Sets Syst.*, vol. 158, no. 19, pp. 2118–2133, 2007.

[79] H. S. Lee, “An optimal algorithm for computing the max-min transitive closure of a fuzzy similarity matrix,” *Fuzzy Sets Syst.*, vol. 123, no. 1, pp. 129–136, 2001.

[80] E. Nasibov, C. Atilgan, M. E. Berberler, and R. Nasiboglu, “Fuzzy joint points based clustering algorithms for large data sets,” *Fuzzy Sets Syst.*, vol. 270, pp. 111–126, 2015.

[81] S. R. Jeffery, M. J. Franklin, and M. Garofalakis, “An adaptive RFID middleware for supporting metaphysical data independence,” *VLDB J.*, vol. 17, no. 2, pp. 265–289, 2008.

[82] S. R. Jeffery, U. C. Berkeley, and M. J. Franklin, “Adaptive Cleaning for RFID Data Streams,” *Vldb*, pp. 163–174, 2006.

[83] N. Khoussainova, M. Balazinska, and D. Suciu, “Probabilistic event extraction from RFID data,” *Proc. - Int. Conf. Data Eng.*, pp. 1480–1482, 2008.

[84] J. Rao and S. Doraiswamy, “A deferred cleansing method for RFID data analytics,” *… Very large data bases*, pp. 175–186, 2006.

[85] Y. Wu, H. Shen, and Q. Z. Sheng, “A Cloud-Friendly RFID Trajectory Clustering Algorithm in Uncertain Environments,” *Ieee Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2075–2088, 2015.

[86] M. Nanni and D. Pedreschi, “Time-focused clustering of trajectories of moving objects,” *JIIS*

Spacial Issue Min. Spat. Data, vol. 27, no. 3, pp. 267–268, 2006.

[87] Y. Cheng, S. Huang, T. Lv, and G. Liu, "A New Data Clustering Algorithm," *2010 Fifth Int. Conf. Internet Comput. Sci. Eng.*, pp. 106–111, 2010.

[88] R. J. Kuo, Y. D. Huang, C.-C. Lin, Y.-H. Wu, and F. E. Zulvia, "Automatic kernel clustering with bee colony optimization algorithm," *Inf. Sci. (Ny).*, vol. 283, pp. 107–122, 2014.

[89] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.

[90] S. Das, A. Abraham, and A. Konar, "Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm," *Pattern Recognit. Lett.*, vol. 29, no. 5, pp. 688–699, 2008.

[91] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 332–344, 2006.

[92] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Inf. Sci. (Ny).*, vol. 195, pp. 124–140, 2012.