



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΥΦΥΗΣ ΠΡΟΔΡΑΣΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΔΕΔΟΜΕΝΩΝ
ΜΕΣΑ ΑΠΟ ΡΟΕΣ ΕΡΩΤΗΜΑΤΩΝ

ΠΕΛΟΠΙΔΑΣ ΑΛΕΞΟΠΟΥΛΟΣ
ΕΜΜΑΝΟΥΗΛ ΙΩΑΝΝΗΣ ΔΟΝΔΩΡΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κωνσταντίνος Κολομβάτσος

Αναπληρωτής Καθηγητής

Τμήματος Πληροφορικής & Τηλεπικοινωνιών Πανεπιστημίου Θεσσαλίας

Λαμία έτος 2024



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΥΦΥΗΣ ΠΡΟΔΡΑΣΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ
ΔΕΔΟΜΕΝΩΝ ΜΕΣΑ ΑΠΟ ΡΟΕΣ ΕΡΩΤΗΜΑΤΩΝ

ΠΕΛΟΠΙΔΑΣ ΑΛΕΞΟΠΟΥΛΟΣ
ΕΜΜΑΝΟΥΗΛ ΙΩΑΝΝΗΣ ΔΟΝΔΩΡΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κωνσταντίνος Κολομβάτσος

Αναπληρωτής Καθηγητής

Τμήματος Πληροφορικής & Τηλεπικοινωνιών Πανεπιστημίου Θεσσαλίας

Λαμία έτος 2024



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

INTELLIGENT PROACTIVE DATA
RECOGNITION BETWEEN QUERRY
STREAMS

PELOPIDAS ALEXOPOULOS
EMMANOUIL IOANNHS DONDOROS

FINAL THESIS

ADVISOR

Konstantinos Kolomvatsos

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS
UNIVERSITY OF THESSALY
Lamia year 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 27/10/2024

Ο – Η Δηλ.

Δόντωρος Εμμανουήλ Ιωάννης

Πελοπίδας Αλεξόπουλος

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Στη σύγχρονη εποχή της ανάλυσης και επεξεργασίας δεδομένων, υπάρχει μεγάλη ανάγκη για προδραστικά συστήματα αναγνώρισης τα οποία προσφέρουν σημαντικά πλεονεκτήματα σε καταστάσεις όπου χρήζουν μέγιστης σημασίας η ταχεία λήψη αποφάσεων, τέτοια συστήματα εδραιώνονται στον κυβερνοχώρο και την οικονομική ανάλυση. Με την βοήθεια της μηχανικής μάθησης, συμπεριλαμβανομένης της επιβλεπομένης και μη επιβλεπομένης μάθησης θέτονται οι βάσεις για την ευφυή αναγνώριση. Εμβαθύνοντας ο υπολογισμός ακμών αναφέρεται ως λύση για την επεξεργασία δεδομένων στις παρυφές του δικτύου, επιτρέποντας έτσι αποκρίσεις χαμηλής καθυστέρησης σε πραγματικό χρόνο. Το μοντέλο που έχει προταθεί παρουσιάζει έναν προηγμένο μηχανισμό για την προδραστική αναγνώριση ερωτημάτων, το οποίο συνδυάζει αλγόριθμους συστάδων μη επιβλεπομένης μάθησης με διαχείριση δεδομένων σε πραγματικό χρόνο, αξιολογώντας την απόδοση του μοντέλου μέσα από μια σειρά πειραμάτων, αποδεικνύοντας την ικανότητα του να προσαρμόζεται δυναμικά στις μεταβαλλόμενες ροές δεδομένων και να εντοπίζει επιτυχώς μοτίβα. Παρουσιάζουμε το υπό εξέταση πρόβλημα και προτείνουμε μια λύση για αυτό, ενώ, ταυτόχρονα αποκαλύπτουμε τα πλεονεκτήματα του μέσα από έναν μεγάλο αριθμό πειραμάτων.

ABSTRACT

In the modern era of data analysis and processing, there is a need for proactive recognition systems that offer significant advantages in situations where rapid decision-making is of utmost importance, such systems are established in cyberspace and financial analysis. With the help of machine learning, including supervised and unsupervised learning, the foundations for intelligent recognition are laid. Deepening Edge Computing is referred to as a solution for processing data at the edges of network, thus enabling low-latency real-time responses. The proposed model presents an advanced mechanism for proactive query recognition, that combines unsupervised learning clustering algorithms with real-time data management, evaluating the model's performance through a series of experiments, demonstrating its ability to dynamically adapt to changing data streams and successfully identify patterns. We present the problem under consideration and propose a solution for it, while simultaneously revealing its advantages through a large number of experiments.

Table of Contents

ΠΕΡΙΛΗΨΗ	1
ABSTRACT	3
ΚΕΦΑΛΑΙΟ 1 Εισαγωγή	8
(1.1 Εισαγωγή)	8
(1.2 Παράδειγμα προδραστικού υπολογισμού).....	8
(1.3 Κίνητρο)	9
(1.4 Δομή εργασίας).....	9
ΚΕΦΑΛΑΙΟ 2 Μηχανική μάθηση	10
(2.1 Μηχανική μάθηση).....	10
(2.2 Ιστορική αναδρομή)	10
(2.3 Παράδειγμα)	11
(2.4 Επιβλεπόμενη μάθηση)	11
(2.4.1 Κατηγοριοποίηση).....	12
(2.4.2 Παλινδρόμηση).....	12
(2.5 Μη επιβλεπόμενη μάθηση)	13
(2.5.1 Συσταδοποίηση).....	14
(2.5.2 Μετασχηματισμός)	15
(2.6 Ενισχυτική μάθηση).....	16
(2.7 Αξιολόγηση των μοντέλων)	17
(2.7.1 Ορθότητα).....	18
(2.7.2 Ακρίβεια και ανάκληση).....	18
(2.7.3 Βαθμολογία F1).....	19
(2.7.4 Χαρακτηριστικό λειτουργίας δέκτη και περιοχή κάτω από την καμπύλη)	20
(2.7.4 Απώλεια καταγραφής)	21
ΚΕΦΑΛΑΙΟ 3 Διαχείριση δεδομένων στις παρυφές του δικτύου	23
(3.1 Internet of Things).....	23
(3.2 Cloud computing)	24
(3.3 Edge computing)	25
(3.4 Χαρακτηριστικά Edge computing)	26
(3.4.1 Πυκνή γεωγραφική κατανομή)	26
(3.4.2 Υποστήριξη κινητικότητας)	26

(3.4.3 Επίγνωση τοποθεσίας).....	26
(3.4.4 Εγγύτητα)	26
(3.4.5 Χαμηλή καθυστέρηση)	27
(3.4.6 Επίγνωση του προβλήματος)	27
(3.4.7 Ανομοιογένεια).....	27
(3.5 Πλεονεκτήματα Edge computing).....	28
(3.6 Εφαρμογές του IoT που βασίζονται σε Edge).....	29
ΚΕΦΑΛΑΙΟ 4 Προτεινόμενο μοντέλο.....	32
(4.1 Gaussian distribution)	32
(4.2 Αφαίρεση ακραίων τιμών με την χρήση του Z-score)	33
(4.3 Αλγόριθμοι συσταδοποίησης που χρησιμοποιήθηκαν)	34
(4.3.1 Αθροιστική συσταδοποίηση-Agglomerative Clustering).....	35
(4.3.2 DBSCAN- Χωρική Ομαδοποίηση Εφαρμογών με Θόρυβο με Βάση την Πυκνότητα)	35
(4.3.3 OPTICS).....	37
(4.4 Συνδυασμός και διαχείριση επικαλυπτόμενων συστάδων)	38
(4.5 Υπολογισμός ποσοστού επιτυχίας μεταξύ κόμβων).....	38
ΚΕΦΑΛΑΙΟ 5 Πειραματική αποτίμηση	39
(5.1 Σύνολο δεδομένων που χρησιμοποιήθηκε)	39
(5.2 Παράμετροι πειραμάτων).....	41
(5.3 Αποτελέσματα και αξιολόγηση πειραμάτων)	42
(5.3.1 Πειράματα με DBSCAN).....	42
(5.3.2 Πειράματα με Agglomerative)	44
(5.3.3 Πειράματα με OPTICS)	46
(5.3.4 Υπολογισμός ποσοστού επιτυχίας)	48
(5. 3.5 Ποσοστό επιτυχίας μεταξύ κόμβων)	49
ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα και Μελλοντικές Προεκτάσεις.....	52
(6.1 Συμπεράσματα)	52
(6.2 Μελλοντικές Προεκτάσεις).....	52
(6.2.1 Επεκτασιμότητα και κατανεμημένη επεξεργασία).....	52
(6.2.2 Προσαρμοζόμενα Συστήματα Μάθησης)	53
Παράρτημα Κώδικες.....	54
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	61

EIKONEΣ	64
----------------------	-----------

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

(1.1 Εισαγωγή)

Οι τεχνολογίες ανάλυσης και επεξεργασίας δεδομένων εξελίσσονται ταχύρρυθμα στην σύγχρονη εποχή, προσφέροντας νέα εργαλεία για την επίλυση προβλημάτων. Η ευφυής προδραστική αναγνώριση ερωτημάτων μέσα από ροές δεδομένων είναι μία από αυτές τις τεχνολογίες, συνδυάζοντας την τεχνητή νοημοσύνη με την επεξεργασία μεγάλων όγκων δεδομένων σε πραγματικό χρόνο. Ο προδραστικός σχεδιασμός είναι μια μέθοδος που ενεργά αναγνωρίζει το πλαίσιο και τις πληροφορίες χρηστών όταν ο στόχος δεν δίνεται ρητά και τελικά σχεδιάζει και εκτελεί προδραστικά το σχέδιο χωρίς παρέμβαση του χρήστη [1],[2]. Ο προδραστικός υπολογισμός [3] προτείνει μια προδραστική διάσταση στον παρόντα υπολογισμό. Ένα σύστημα πρόβλεψης μπορεί να οριστεί ως ένα σύστημα που περιέχει ένα προγνωστικό μοντέλο του εαυτού του ή και του περιβάλλοντος του οποίου του επιτρέπει να αλλάξει κατάσταση στιγμιαία σύμφωνα με τις προβλέψεις του μοντέλου που αντιστοιχούν σε μια μεταγενέστερη στιγμή.

(1.2 Παράδειγμα προδραστικού υπολογισμού)

Ο προδραστικός υπολογισμός, επομένως, εστιάζει στη λειτουργία υπό ανθρώπινη επίβλεψη, όπου ο χρήστης μένει εκτός βρόχου όσο το δυνατόν περισσότερο έως ότου απαντηθεί καθοδήγηση σε κρίσιμες αποφάσεις. Ένα απλό σημερινό παράδειγμα που απεικονίζει ένα σύστημα υπό ανθρώπινη επίβλεψη είναι ένα σύγχρονο σύστημα κεντρικής θέρμανσης. Τέτοια συστήματα έχουν συνήθως ρυθμίσεις θερμοκρασίας πρωί, ημέρα, βράδυ και νύχτα. Κανονικά, το σύστημα λειτουργεί ανεπίβλεπτα και απαρατήρητα. Ωστόσο, ο χρήστης μπορεί εύκολα να παρακάμψει αυτές τις ρυθμίσεις οποιαδήποτε στιγμή, εάν αισθανθεί ζέστη ή κρύο, ή για να απευθυνθεί σε μία επικείμενη ενεργειακή κρίση. Επιπλέον, εάν το σύστημα ήταν εξοπλισμένο με δίκτυο αισθητήρων και γνωρίζοντας το ημερολόγιο μιας οικογένειας, η θερμοκρασία και η κατανάλωση ενέργειας θα μπορούσε να βελτιστοποιηθεί προδραστικά επιτρέποντας ενδοοικογενειακά μικροκλίματα, αργά τις εργάσιμες ημέρες και στις οικογενειακές διακοπές. Ωστόσο, επεκτείνοντας αυτό το παράδειγμα σε πιο περίπλοκα συστήματα είναι μια μεγάλη πρόκληση καθώς οι περισσότερες αποφάσεις δεν καταλήγουν απλώς στο «πολύ ζεστό» ή «πολύ κρύο» [4].

(1.3 Κίνητρο)

Το κίνητρο για την εκπόνηση της παρούσας πτυχιακής εργασίας έγκειται στην ανάγκη για αποτελεσματική και άμεση αναγνώριση και ανάλυση πληροφοριών σε περιβάλλοντα όπου η ταχύτητα λήψης αποφάσεων είναι κρίσιμη. Σε πολλές εφαρμογές, όπως η ασφάλεια στον κυβερνοχώρο, η χρηματοοικονομική ανάλυση και η ιατρική διάγνωση, η δυνατότητα να εντοπίζονται σημαντικά μοτίβα και να λαμβάνονται αποφάσεις προδραστικά μπορεί να οδηγήσει σε σημαντικά πλεονεκτήματα.

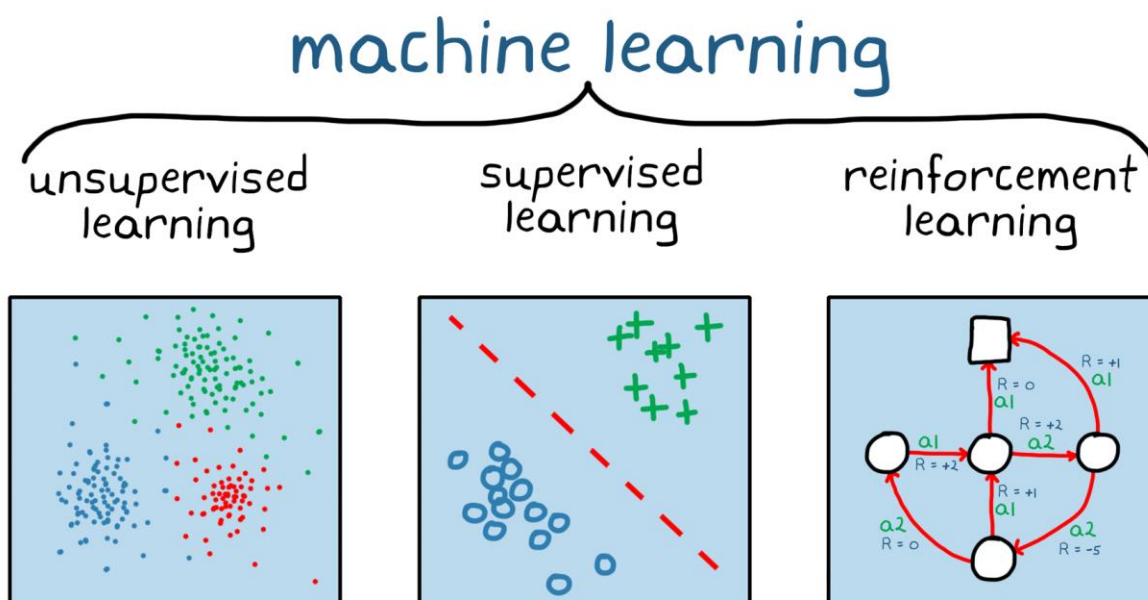
(1.4 Δομή εργασίας)

Η πτυχιακή εργασία δομείται σε 6 κεφάλαια κατά τον ακόλουθο δομημένο τρόπο. Στο κεφάλαιο 2 πραγματοποιείται ανάλυση της έννοιας μηχανικής μάθησης παραθέτοντας ιστορική αναδρομή και παραδείγματα αναφέροντας τις επιμέρους κατηγορίες της, καθώς και πώς αξιολογούνται αυτά τα μοντέλα μηχανικής μάθησης. Στο Κεφάλαιο 3 περιγράφεται η διαχείριση των δεδομένων στις παρυφές του δικτύου. Αναφερόμενοι κυρίως στο Edge Computing παραθέτοντας τα χαρακτηριστικά του, τα πλεονεκτήματα και τις εφαρμογές του στο IoT. Στο κεφάλαιο 4 περιγράφονται αναλυτικά οι αλγόριθμοι συστάδων που χρησιμοποιήθηκαν καθώς και η χρήση της κανονικής κατανομής και της ευκλείδειας απόστασης. Στο κεφάλαιο 5 γίνεται η παρουσίαση και η αξιολόγηση των αποτελεσμάτων του πειράματος. Τέλος, στο κεφάλαιο 6, παραθέτονται τα τελικά συμπεράσματα που προέκυψαν μέσα από την υλοποίηση του πειράματος ακολουθούμενα από μελλοντικές προεκτάσεις πάνω στο θέμα.

ΚΕΦΑΛΑΙΟ 2 Μηχανική μάθηση

(2.1 Μηχανική μάθηση)

Η μηχανική μάθηση είναι ένας κλάδος της τεχνητής νοημοσύνης που δίνει την δυνατότητα στις μηχανές να μπορούν να αντλήσουν και να μάθουν από τα δεδομένα παραδειγμάτων και προηγούμενων εμπειριών, καθώς θα αναγνωρίζουν μοτίβα για να κάνουν προβλέψεις έχοντας την ελάχιστη ανθρώπινη παρέμβαση. Οι μέθοδοι μηχανικής μάθησης επιτρέπουν στους υπολογιστές να λειτουργούν αυτόνομα χωρίς σαφή προγραμματισμό. Οι εφαρμογές μηχανικής μάθησης τροφοδοτούνται με νέα δεδομένα και μέσω αυτών μαθαίνουν αυτόνομα, εξελίσσονται και προσαρμόζονται. Η μηχανική μάθηση αντλεί πληροφορίες από μεγάλους όγκους δεδομένων χρησιμοποιώντας αλγόριθμους για να εντοπίσουν μοτίβα μέσω μιας επαναληπτικής διαδικασίας. Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούν υπολογιστικές μεθόδους για να μάθουν κατευθείαν από τα δεδομένα αντί να βασίζονται σε οποιαδήποτε προκαθορισμένη εξίσωση που μπορεί να χρησιμοποιηθεί ως μοντέλο [5].



Εικόνα 1: Είδη μηχανικής μάθησης

(2.2 Ιστορική αναδρομή)

Οι έρευνες για την τεχνητή νοημοσύνη ξεκίνησαν τη δεκαετία του 1950. Εκείνα τα χρόνια ο Alan Turing εκτέλεσε το τεστ Turing για να δοκιμάσει την ικανότητα μιας μηχανής να μιμείται έναν άνθρωπο. Ο στόχος της δοκιμής Turing ήταν να μετρήσει

την ικανότητα της μηχανής να επικοινωνήσει με έναν άνθρωπο κατά τη διάρκεια μιας συνέντευξης. Αν η μηχανή αποδώσει χειρότερα από έναν άνθρωπο, θεωρείται επιτυχημένη. Το 1956, ο όρος «τεχνητή νοημοσύνη» χρησιμοποιήθηκε για πρώτη φορά σε ένα θερινό σχολείο που πραγματοποιήθηκε από τον Marvin Minsky από το Ινστιτούτο Τεχνολογίας της Μασαχουσέτης, John McCarthy από το Stanford University και τους Allen Newell και Herbert Simon από το Carnegie-Mellon University. Το 1959, ο Arthur Samuel δημιούργησε το πρόγραμμα ντάμας και τότε η μηχανική μάθηση άρχισε να αναπτύσσεται. Από αυτές τις εξελίξεις μέχρι τη δεκαετία του 1980, υπήρξαν μελέτες για την αφηρημένη νόηση και τα συστήματα βασισμένα σε πληροφορίες, τα οποία ονομάστηκαν "χειμώνας της τεχνητής νοημοσύνης". Στη δεκαετία του 1990, οι μελέτες σχετικά με την τεχνητή νοημοσύνη και τη μηχανική μάθηση επιταχύνθηκαν λόγω των εξελίξεων στις τεχνολογίες παιχνιδιών. Σήμερα, η τεχνητή νοημοσύνη και η μηχανική μάθηση χρησιμοποιούνται σε πολλές έρευνες και τομείς εργασίας [6].

(2.3 Παράδειγμα)

Ένα παράδειγμα για την κατανόηση της λειτουργίας της μηχανικής μάθησης είναι η δυναμική τιμολόγηση η οποία επιτρέπει στις εταιρείες να προσαρμόζουν τις τιμές που χρεώνουν τα προϊόντα και τις υπηρεσίες τους σε σχεδόν πραγματικό χρόνο με βάση τις μεταβαλλόμενες συνθήκες της αγοράς. "Εξετάζετε τη συμπεριφορά των καταναλωτών και τα μοτίβα αγορών για να αυξομειώσετε τις τιμές σας. Είναι μια πολύ πολύτιμη χρήση της μηχανικής μάθησης για τις εταιρείες", δήλωσε ο Vikas Agarwal, επικεφαλής σε θέματα χρηματοοικονομικών υπηρεσιών και ρυθμιστικών αρχών στην εταιρεία επαγγελματικών υπηρεσιών PwC. Τα συστήματα μηχανικής μάθησης συνήθως χρησιμοποιούν πολυάριθμα σύνολα δεδομένων, όπως μακροοικονομικά δεδομένα και δεδομένα μέσω κοινωνικής δικτύωσης, για να ορίσουν και να επαναφέρουν τις τιμές. Αυτό γίνεται συνήθως για αεροπορικά εισιτήρια, τιμές δωματίων ξενοδοχείων και κόμιστρα διαμοιρασμένης διαδρομής. Η αυξανόμενη τιμολόγηση της Uber, όπου οι τιμές αυξάνονται όταν αυξάνεται η ζήτηση, είναι ένα χαρακτηριστικό παράδειγμα του τρόπου με τον οποίο οι εταιρείες χρησιμοποιούν αλγόριθμους μηχανικής μάθησης για να προσαρμόσουν τις τιμές καθώς αλλάζουν οι συνθήκες [7].

(2.4 Επιβλεπόμενη μάθηση)

Οι επιβλεπόμενες τεχνικές μηχανικής μάθησης αξιοποιούν την γνώση που έχουν αποκτήσει από τα προηγούμενα και τρέχοντα δεδομένα με την βοήθεια ετικετών ώστε να προβλέψουν γεγονότα. Αυτή η προσέγγιση ξεκινάει από την διαδικασία εκπαίδευσης του συνόλου των δεδομένων όπου η μηχανική μάθηση αναπτύσσει μια εξαγόμενη λειτουργία για να προβλέψει τις τιμές εξόδου. Το σύστημα είναι ικανό να παρέχει

αποτελέσματα σε δεδομένα εισόδου μετά από επαρκή εκπαίδευση. Ο αλγόριθμος μηχανικής μάθησης συγκρίνει τα ληφθέντα αποτελέσματα με τα πραγματικά και αναμενόμενα αποτελέσματα ώστε να αναγνωρίσει τα σφάλματα για να τροποποιήσει το μοντέλο βασισμένο στα αποτελέσματα [8].

(2.4.1 Κατηγοριοποίηση)

Η κατηγοριοποίηση(classification) είναι ένας επιβλεπόμενος αλγόριθμος μηχανικής μάθησης που προβλέπει μια κατηγορική ή διακριτή μεταβλητή εξόδου με βάση τις μεταβλητές εισόδου. Οι μεταβλητές εισόδου συχνά ονομάζονται χαρακτηριστικά ή προγνωστικά, ενώ η μεταβλητή εξόδου ονομάζεται κατηγορία ή ετικέτα. Ένας αλγόριθμος ταξινόμησης στοχεύει να μάθει μια συνάρτηση χαρτογράφησης από τις εισόδους στις εξόδους με βάση ένα επισημασμένο σύνολο εκπαίδευσης. Το σύνολο εκπαίδευσης αποτελείται από περιπτώσεις, καθεμία από τις οποίες σχετίζεται με μια ετικέτα κατηγορίας. Ο αλγόριθμος χρησιμοποιεί αυτά τα επισημασμένα δεδομένα για να δημιουργήσει ένα μοντέλο που προβλέπει την ετικέτα κατηγορίας νέων, άγνωστων περιπτώσεων με βάση τις μεταβλητές εισόδου τους. Υπάρχουν αρκετοί αλγόριθμοι κατηγοριοποίησης, όπως τα δέντρα απόφασης (decision trees), τα τυχαία δάση (random forests), η λογιστική παλινδρόμηση (logistic regression) και οι μηχανές υποστήριξης διανυσμάτων (SVMs).

Οι αλγόριθμοι κατηγοριοποίησης έχουν πολλές εφαρμογές, όπως η αναγνώριση εικόνων, η επεξεργασία φυσικής γλώσσας, η ανίχνευση απάτης και η αξιολόγηση πιστοληπτικής ικανότητας [9].

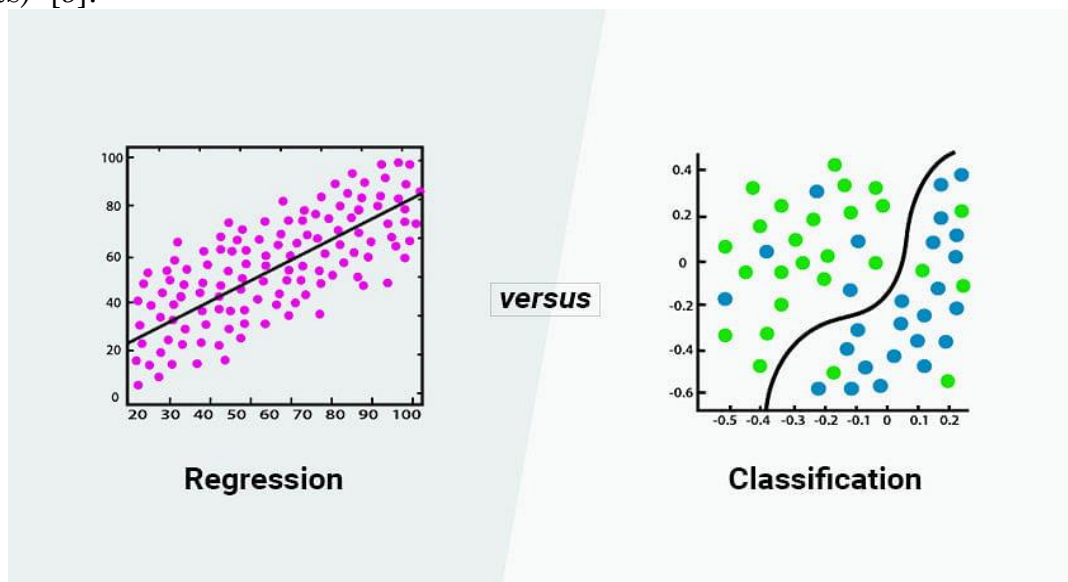
(2.4.2 Παλινδρόμηση)

Η παλινδρόμηση (Regression) [9] είναι ένας επιβλεπόμενος αλγόριθμος μηχανικής μάθησης που προβλέπει μια συνεχή αριθμητική μεταβλητή εξόδου με βάση τις μεταβλητές εισόδου. Οι μεταβλητές εισόδου συχνά ονομάζονται χαρακτηριστικά ή προγνωστικά, ενώ η μεταβλητή εξόδου ονομάζεται στόχος ή εξαρτημένη μεταβλητή.

Ένας αλγόριθμος παλινδρόμησης στοχεύει να μάθει μια συνάρτηση χαρτογράφησης από τις εισόδους στις εξόδους με βάση ένα επισημασμένο σύνολο εκπαίδευσης. Το σύνολο εκπαίδευσης αποτελείται από περιπτώσεις που συνδέονται με μια αριθμητική τιμή στόχου. Ο αλγόριθμος χρησιμοποιεί αυτά τα επισημασμένα δεδομένα για να δημιουργήσει ένα μοντέλο που προβλέπει την τιμή στόχου νέων, άγνωστων περιπτώσεων με βάση τις μεταβλητές εισόδου του.

Υπάρχουν πολλοί αλγόριθμοι παλινδρόμησης, όπως η γραμμική παλινδρόμηση (linear regression), η πολυωνυμική παλινδρόμηση (polynomial regression), η παλινδρόμηση

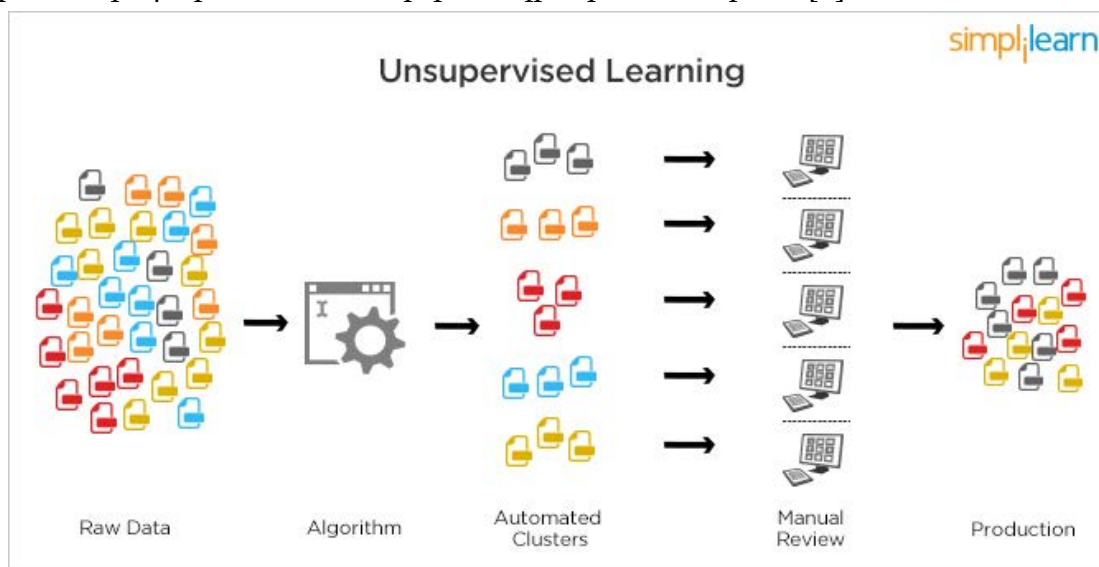
με δέντρα απόφασης (decision trees) και η παλινδρόμηση με τυχαία δάση (random forests) [9].



Εικόνα 2: Παλινδρόμηση / Κατηγοριοποίηση

(2.5 Μη επιβλεπόμενη μάθηση)

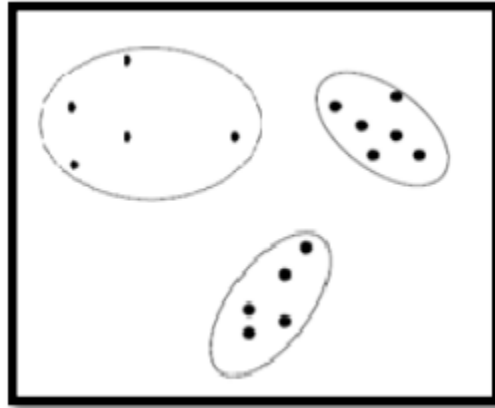
Οι μη επιβλεπόμενες τεχνικές μηχανικής μάθησης χρησιμοποιούνται όταν τα δεδομένα εκπαίδευσης δεν είναι ταξινομημένα και δεν υπάρχουν ετικέτες. Αναλύουν πως το σύστημα μπορεί να εξάγει μια συνάρτηση για να εξηγήσει τα κρυφά μοτίβα από τα μη επισημασμένα μοντέλα. Το σύστημα δεν προσδιορίζει την κατάλληλη έξοδο, αλλά ανακαλύπτει τα δεδομένα και καταγράφει παρατηρήσεις από το σύνολο δεδομένων για να βρει τα κρυφά μοτίβα από τα μη επισημασμένα δεδομένα [8].



Εικόνα 3: Μη επιβλεπόμενη μάθηση

(2.5.1 Συσταδοποίηση)

Η ανάλυση συστάδων(clusters) είναι μια διερευνητική τεχνική της οποίας ο στόχος είναι να προσδιορίσει ομάδες, ή συστάδες υψηλής πυκνότητας στις οποίες οι παρατηρήσεις μοιάζουν περισσότερο η μια με την άλλη παρά με τις παρατηρήσεις που έχουν εκχωρηθεί σε διαφορετικές συστάδες. Αυτή η διαδικασία απαιτεί να ποσοτικοποιηθεί ο βαθμός ομοιότητας ή ανομοιότητας, μεταξύ των παρατηρήσεων. Τα αποτελέσματα της ανάλυσης εξαρτώνται σε μεγάλο βαθμό από το είδος της χρησιμοποιούμενης μέτρησης ομοιότητας.



Εικόνα 4: Συσταδοποίηση

Μια μεγάλη κατηγορία ανομοιοτήτων συμπίπτει με την κατηγορία των συναρτήσεων απόστασης. Η πιο χρησιμοποιούμενη συνάρτηση απόστασης είναι η Ευκλείδεια απόσταση, όπως ορίζεται το άθροισμα των τετραγωνικών διαφορών των χαρακτηριστικών μεταξύ δύο μοτίβων x_i και x_j

$$d(x_i, x_j) = \sum_{h=1}^p (x_{ih} - x_{jh})^2$$

όπου x_{ih} είναι το h-ιοστο χαρακτηριστικό του x_i .

Ομοίως, οι αλγόριθμοι συσταδοποίηση μπορούν να χρησιμοποιήσουν επίσης μέτρα ομοιότητας μεταξύ τους παρατηρήσεις. Ο συντελεστής συσχέτισης χρησιμοποιείται

$$\text{ευρέως ως μέτρο ομοιότητας : } p(x_i, x_j) = \frac{\sum_h (x_{ih} - \bar{x}_i)(x_{jh} - \bar{x}_j)}{\sum_h (x_{ih} - \bar{x}_i)^2 \sum_h (x_{jh} - \bar{x}_j)^2}$$

$$\text{Όπου } \bar{x}_i = \frac{1}{p} \sum_{h=1}^p x_{ih}$$

είναι η μέση τιμή των χαρακτηριστικών μιας ενιαία παρατήρησης [10].

Οι αλγόριθμοι συσταδοποίησης ταξινομούνται σε δύο κατηγορίες: ιεραρχική συσταδοποίηση και τμηματική συσταδοποίηση.

Τύποι συστάδων :

- Καλά διαχωρισμένες συστάδες(Well-separated clusters):
Μια συστάδα μπορεί να αποτελείται από ένα σύνολο σημείων, επομένως οποιοδήποτε σημείο σε ένα σύνολο είναι πιο κοντά σε κάθε διαφορετικό σημείο μέσα στη συστάδα σε σύγκριση με κάθε άλλο σημείο που δεν είναι μέσα στη συστάδα.
- Συστάδες που βασίζονται στο κέντρο(Center-based clusters) :
Μια συστάδα θα μπορούσε να είναι ένα σύνολο αντικειμένων, επομένως ένα αντικείμενο μέσα σε ένα σύνολο να είναι πιο κοντά στο «κέντρο» της συστάδας από ό,τι στο κέντρο οποιουδήποτε άλλης συστάδας. Το κέντρο μιας συστάδας είναι συνήθως ένα κεντροειδές.
- Συνεχείς συστάδες(Contiguous clusters) :
Μια συστάδα μπορεί να είναι μια θέση σημείων, επομένως οποιοδήποτε σημείο μέσα σε ένα σύνολο είναι πιο κοντά με ένα ή περισσότερα άλλα σημεία μέσα στη συστάδα σε σύγκριση με οποιοδήποτε σημείο που δεν βρίσκεται μέσα στη συστάδα.
- Συστάδες με βάση την πυκνότητα(Density-based clusters) :
Μια συστάδα μπορεί να είναι μια πυκνή περιοχή σημείων που χωρίζεται από άλλες περιοχές υψηλής πυκνότητας σύμφωνα με τις περιοχές χαμηλής πυκνότητας [11].

(2.5.2 Μετασχηματισμός)

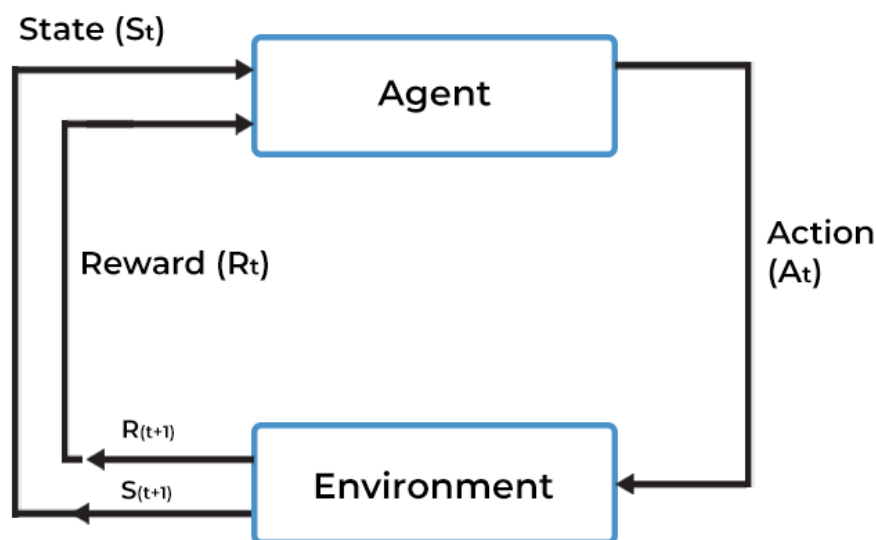
Προηγούμενες έρευνες έχουν δείξει ότι οι χώροι δεδομένων υψηλών διαστάσεων είναι ως επί το πλείστον κενές, υποδεικνύοντας ότι η δομή δεδομένων που εμπλέκεται υπάρχει κυρίως σε έναν υποχώρο. Ως αποτέλεσμα, υπάρχει ανάγκη για μεθόδους εξαγωγής χαρακτηριστικών οι οποίες θα μπορούν να μειώσουν τη διάσταση των δεδομένων σε αυτόν τον υποχώρο, διατηρώντας παράλληλα τις αρχικές πληροφορίες που επιτρέπουν των διαχωρισμό των τάξεων. Σε άλλα λόγια, η μείωση των διαστάσεων είναι ο μετασχηματισμός που φέρνει τα δεδομένα από έναν χώρο υψηλότερης διάστασης σε έναν χώρο χαμηλότερης, ξεπερνώντας έτσι την «κατάρρα» της διάστασης. Μία από τις πιο ευρέως χρησιμοποιούμενες τεχνικές μείωσης διαστάσεων στην τηλεπισκόπηση είναι η ανάλυση του κύριου στοιχείου (PCA- Principal Component Analysis). Το PCA υπολογίζει τις ορθογώνιες προβολές που μεγιστοποιούν την ποσότητα διακύμανσης των δεδομένων και αποδίδει ένα σύνολο δεδομένων σε ένα ασύνδετο σύστημα συντεταγμένων [12].

(2.6 Ενισχυτική μάθηση)

Η ενισχυτική μάθηση (reinforcement learning) είναι το πρόβλημα που αντιμετωπίζει ένας πράκτορας που πρέπει να μάθει συμπεριφορά μέσω αλληλεπιδράσεων δοκιμής και σφάλματος με ένα δυναμικό περιβάλλον. Υπάρχει μια ισχυρή οικογενειακή ομοιότητα με την ομώνυμη εργασία στην ψυχολογία, αλλά διαφέρει σημαντικά στις λεπτομέρειες και στη χρήση της λέξης ενίσχυσης. Θεωρείται κατάλληλα ως μια κατηγορία προβλημάτων παρά ως ένα σύνολο τεχνικών. Υπάρχουν δύο κύριες στρατηγικές για την επίλυση των προβλημάτων ενισχυτικής μάθησης. Η πρώτη είναι η αναζήτηση στο χώρο των συμπεριφορών προκειμένου να βρεθεί μία συμπεριφορά που έχει καλή απόδοση στο περιβάλλον. Αυτή η προσέγγιση έχει υιοθετηθεί με την εργασία γενετικών αλγορίθμων και γενετικού προγραμματισμού, καθώς και με μερικές ακόμη καινοτόμες τεχνικές αναζήτησης. Η δεύτερη είναι η χρήση στατιστικών τεχνικών και μεθόδων δυναμικού προγραμματισμού για την εκτίμηση της χρησιμότητας της λήψης δράσεων σε καταστάσεις του κόσμου [13].



REINFORCEMENT LEARNING MODEL



Εικόνα 5: Αλληλεπίδραση Πράκτορα-Περιβάλλον στην Ενισχυτική μάθηση

Στην εικόνα 5 παρουσιάζονται όλες οι αλληλεπιδράσεις του πράκτορα με το περιβάλλον. Συγκεκριμένα, ο πράκτορας τοποθετείτε σε ένα σχεδιασμένο περιβάλλον με πιθανές καταστάσεις (State), με κάθε ενέργεια του πράκτορα να του αποδίδετε μια ανταμοιβή (Reward). Ένα παράδειγμα ενισχυτικής μάθησης αποτελούν τα επιτραπέζια

παιχνίδια, διότι με τις ενέργειες που θα ακολουθήσει θα του δίνεται η αντίστοιχη ανταμοιβή(νίκη ή ήττα).

(2.7 Αξιολόγηση των μοντέλων)

Η αξιολόγηση αποτελεί έναν απαραίτητο ρόλο στη διαδικασία δημιουργίας ισχυρών και αποτελεσματικών μοντέλων κατηγοριοποίησης, ας αναλύσουμε γιατί:

- Κατανόηση της απόδοσης του μοντέλου: Οι μετρήσεις αξιολόγησης παρέχουν μετρήσεις για να κατανοήσουμε πόσο καλά αποδίδει το μοντέλο.
- Επιλογή μοντέλου: Όταν έχουμε πολλά μοντέλα στην διάθεσή μας, οι μετρήσεις αξιολόγησης βοηθούν στην επιλογή του καλύτερου μοντέλου. Μας δίνουν τη δυνατότητα να συγκρίνουμε αντικειμενικά την απόδοση διαφορετικών μοντέλων στο ίδιο πρόβλημα.
- Βελτίωση μοντέλου: Οι μετρήσεις αξιολόγησης μπορούν να βοηθήσουν στον εντοπισμό των περιοχών όπου το μοντέλο υπολειτουργεί. Αναλύοντας αυτές τις περιοχές, μπορούμε να κάνουμε τις απαραίτητες βελτιώσεις ή τροποποιήσεις στο μοντέλο ή στις δυνατότητες που χρησιμοποιούμε.
- Ανίχνευση υπερβολικής προσαρμογής: Η υπερβολική προσαρμογή είναι ένα κοινό πρόβλημα στη μηχανική μάθηση, όπου ένα μοντέλο έχει καλή απόδοση στα δεδομένα εκπαίδευσης, αλλά κακή σε μη ορατά δεδομένα. Η σωστή αξιολόγηση χρησιμοποιώντας ένα σετ επικύρωσης αναμονής ή διασταυρούμενη επικύρωση μπορεί να βοηθήσει στην ανίχνευση υπερβολικής προσαρμογής.
- Μέτρηση επιχειρηματικού αντίκτυπου: Σε πρακτικές εφαρμογές, τα ενδιαφερόμενα μέρη ενδιαφέρονται συχνά για την αναμενόμενη απόδοση ενός μοντέλου. Οι μετρήσεις αξιολόγησης, ειδικά όταν συνδέονται με μετρήσεις επιχειρήσεων, μπορούν να παρέχουν αυτές τις πληροφορίες.

Ουσιαστικά, χωρίς την κατάλληλη αξιολόγηση, δεν μπορεί να εξακριβωθεί εάν έχουν επιτευχθεί οι στόχοι, ούτε μπορούν να παρθούν τεκμηριωμένες αποφάσεις σχετικά με την περαιτέρω ανάπτυξη του μοντέλου.

Στις εργασίες κατηγοριοποίησης, υπάρχει μια ποικιλία διαθέσιμων μετρήσεων για την αξιολόγηση της απόδοσης του μοντέλου. Κάθε μέτρηση παρέχει μια μοναδική προοπτική για την ικανότητα του μοντέλου, προσφέροντας πληροφορίες για διαφορετικές πτυχές των αποτελεσμάτων. από Οι μετρήσεις αυτές είναι οι εξής:

(2.7.1 Ορθότητα)

Η ορθότητα είναι μια από τις πιο απλές μετρήσεις που χρησιμοποιούνται στη μηχανική μάθηση. Παρέχει ένα απλό, μέτρο της συνολικής απόδοσης ενός μοντέλου. Στο πλαίσιο της κατηγοριοποίησης, η ορθότητα είναι το ποσοστό των σωστών προβλέψεων που γίνονται από το μοντέλο από όλες τις προβλέψεις. Μαθηματικά, η ορθότητα υπολογίζεται ως ο αριθμός των σωστών προβλέψεων (τόσο των αληθινών θετικών όσο και των αληθινών αρνητικών) διαιρεμένος με τον συνολικό αριθμό των προβλέψεων (το άθροισμα των αληθινών θετικών, των αληθινών αρνητικών, των ψευδώς θετικών και των ψευδών αρνητικών). Μαθηματικά, η ακρίβεια μπορεί να εκφραστεί χρησιμοποιώντας τα στοιχεία του πίνακα σύγχυσης: True Positives (TP), True Negatives (TN), False Positives (FP) και False Negatives (FN).

Ο τύπος για την ακρίβεια είναι:

$$\text{Ακρίβεια} = \frac{TP + TN}{TP + TN + FP + FN}$$

Όπου:

True Positives (TP): Οι περιπτώσεις στις οποίες το μοντέλο προέβλεψε σωστά τη θετική τάξη.

True Negatives (TN): Οι περιπτώσεις στις οποίες το μοντέλο προέβλεψε σωστά την αρνητική τάξη.

False Positives (FP): Οι περιπτώσεις στις οποίες το μοντέλο προέβλεψε εσφαλμένα τη θετική κλάση (γνωστό και ως σφάλμα τύπου I).

False Negatives (FN): Οι περιπτώσεις στις οποίες το μοντέλο προέβλεψε εσφαλμένα την αρνητική κλάση (γνωστό και ως σφάλμα τύπου II).

Με απλά λόγια, η ορθότητα μας δίνει το κλάσμα των σωστών προβλέψεων.

(2.7.2 Ακρίβεια και ανάκληση)

Η ακρίβεια και η ανάκληση είναι δύο θεμελιώδη μέτρα που χρησιμοποιούνται στην αξιολόγηση μοντέλων ταξινόμησης, ειδικά σε καταστάσεις όπου τα ψευδώς θετικά και τα ψευδώς αρνητικά έχουν σημαντικές συνέπειες. Με τους απλούστερους όρους, η ακρίβεια παρέχει ένα μέτρο της ακρίβειας ή της ποιότητας ενός μοντέλου. Η υψηλή ακρίβεια υποδηλώνει χαμηλό ποσοστό ψευδώς θετικών προβλέψεων. Από την άλλη πλευρά η ανάκληση παρέχει ένα μέτρο της πληρότητας ή της ποσότητας ενός μοντέλου. Η υψηλή ανάκληση υποδηλώνει χαμηλό ποσοστό ψευδώς αρνητικών προβλέψεων. Αυτές οι μετρήσεις είναι ιδιαίτερα χρήσιμες σε σενάρια όπου ένας τύπος σφάλματος είναι πιο σημαντικό να αποφευχθεί από τον άλλο. Για παράδειγμα, σε ένα μοντέλο

ανίχνευσης ανεπιθύμητων μηνυμάτων, μπορεί να θέλουμε να δώσουμε προτεραιότητα στην ακρίβεια για να αποφύγουμε την επισήμανση νόμιμων μηνυμάτων ηλεκτρονικού ταχυδρομείου ως ανεπιθύμητων (ψευδώς θετικά). Αντίθετα, σε ένα μοντέλο διάγνωσης ασθενειών, θα μπορούσαμε να δώσουμε προτεραιότητα στην ανάκληση για να διασφαλίσουμε ότι δεν θα χάσουμε καμία θετική περίπτωση (ψευδώς αρνητικά). Ο τύπος για την ανάκληση (επίσης γνωστός ως ευαισθησία ή αληθινός θετικός ρυθμός) είναι:

$$\text{Ανάκληση} = \frac{TP}{TP + FP}$$

Είναι σημαντικό να σημειωθεί ότι η ακρίβεια και η ανάκληση σχετίζονται αντιστρόφως. Καθώς αυξάνεται η ακρίβεια, η ανάκληση πέφτει και αντίστροφα. Αυτή η αντιστάθμιση μας οδηγεί σε μια άλλη σημαντική μέτρηση γνωστή ως βαθμολογία F1.

(2.7.3 Βαθμολογία F1)

Ενώ η ακρίβεια και η ανάκληση είναι απίστευτα ενημερωτικές, μερικές φορές παρέχουν μια μερική άποψη της απόδοσης ενός μοντέλου. Όπως αναφέρθηκε, τα μοντέλα αντιμετωπίζουν συχνά μια αντιστάθμιση μεταξύ ακρίβειας και ανάκλησης. Το ένα μπορεί να είναι υψηλό ενώ το άλλο χαμηλό, κάτι που μπορεί να είναι προβληματικό όταν το μοντέλο χρειάζεται να έχει ισορροπία και των δύο. Με την βαθμολογία F1 παρέχεται μια ισορροπία μεταξύ ακρίβειας και ανάκλησης και μπορεί να είναι καλύτερο μέτρο από την ακρίβεια. Ορίζεται ως ο αρμονικός μέσος όρος ακρίβειας και ανάκλησης και τους αποδίδεται ίσο βάρος. Σε αντίθεση με τον αριθμητικό μέσο όρο, ο αρμονικός μέσος όρος τείνει προς το μικρότερο από τα δύο στοιχεία. Επομένως, ένα μοντέλο θα έχει υψηλή βαθμολογία F1 μόνο εάν τόσο η ακρίβεια όσο και η ανάκληση είναι υψηλή. Ο μαθηματικός τύπος για τη βαθμολογία F1 είναι:

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{ακρίβεια}} + \frac{1}{\text{ανάκληση}}}$$

$$\text{Απλοποιημένα : F1 Score} = 2 * \frac{\text{ακρίβεια} * \text{ανάκληση}}{\text{ακρίβεια} + \text{ανάκληση}}$$

Χρησιμοποιώντας τον αρμονικό μέσο όρο αντί για τον αριθμητικό μέσο όρο, η βαθμολογία F1 τείνει προς το μικρότερο μεταξύ της ακρίβειας και της ανάκλησης. Αυτό σημαίνει ότι η βαθμολογία F1 θα είναι χαμηλή εάν είτε η ακρίβεια είτε η ανάκληση είναι χαμηλή.

(2.7.4 Χαρακτηριστικό λειτουργίας δέκτη και περιοχή κάτω από την καμπύλη)

Στη μηχανική μάθηση, και ειδικά σε εργασίες δυαδικής κατηγοριοποίησης, η καμπύλη χαρακτηριστικών λειτουργίας δέκτη ή καμπύλη ROC και η περιοχή κάτω από την καμπύλη ή AUC είναι δημοφιλή εργαλεία που χρησιμοποιούνται για την αξιολόγηση της απόδοσης ενός μοντέλου. Αυτές οι μετρήσεις παρέχουν μια ολοκληρωμένη εικόνα της απόδοσης του μοντέλου σε διάφορα όρια κατηγοριοποίησης.

Καμπύλη ROC: Η καμπύλη ROC είναι μια γραφική παράσταση που εμφανίζει τον πραγματικό θετικό ρυθμό (TPR-True Positive Rate) έναντι του ψευδώς θετικού ποσοστού (FPR-False Positive Rate) σε διάφορες ρυθμίσεις ορίου. Κάθε σημείο στην καμπύλη ROC αντιπροσωπεύει ένα ζεύγος ευαισθησίας και ειδικότητας που αντιστοιχεί σε ένα συγκεκριμένο όριο απόφασης. Μια δοκιμή με τέλεια διάκριση (χωρίς επικάλυψη στις δύο κατανομές) έχει μια καμπύλη ROC που διέρχεται από την επάνω αριστερή γωνία (100% ευαισθησία, 100% ειδικότητα). Επομένως, όσο πιο κοντά είναι η καμπύλη ROC στην επάνω αριστερή γωνία, τόσο μεγαλύτερη είναι η συνολική ακρίβεια της δοκιμής.

Για την κατανόηση της καμπύλη ROC, πρέπει να ορίσουμε δύο όρους: True Positive Rate (TPR) και False Positive Rate (FPR).

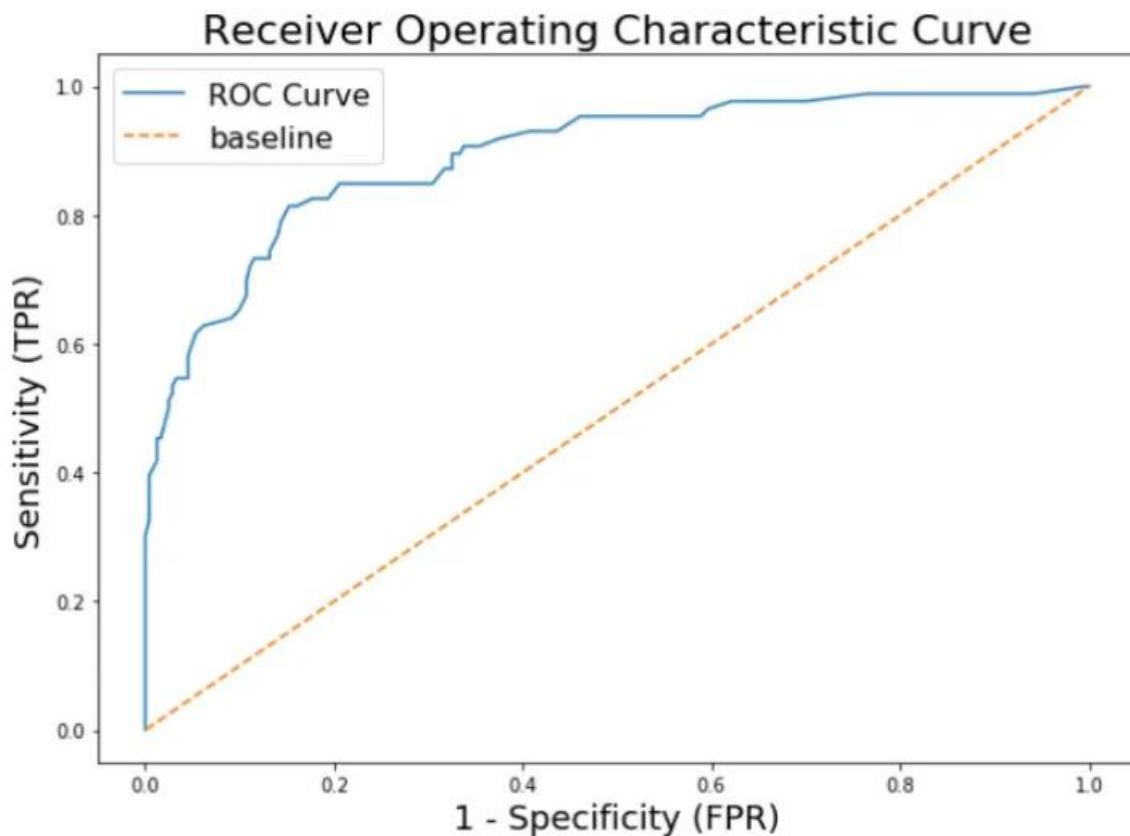
True Positive Rate (TPR): γνωστό ως ευαισθησία ή ανάκληση. Μετρά την αναλογία των πραγματικών θετικών που προσδιορίζονται σωστά ως θετικά. Ο τύπος είναι:

$$TPR = \frac{TP}{TP+FN}$$

False Positive Rate (FPR): μετρά την αναλογία των πραγματικών αρνητικών που εσφαλμένα προσδιορίζονται ως θετικά. Ο τύπος είναι:

$$FPR = \frac{FP}{FP+TN}$$

Η καμπύλη ROC είναι τότε μια γραφική παράσταση TPR (στον άξονα y) έναντι FPR (στον άξονα x) για διαφορετικές τιμές ορίου.



Εικόνα 6: Καμπύλη ROC

AUC: Το AUC σημαίνει περιοχή κάτω από την καμπύλη ROC. Δηλαδή, μετρά ολόκληρη τη διδιάστατη περιοχή κάτω από ολόκληρη την καμπύλη ROC από (0,0) έως (1,1). Η AUC παρέχει ένα συνολικό μέτρο απόδοσης σε όλα τα πιθανά όρια ταξινόμησης. Μια AUC με αριθμητική τιμή 1 υποδηλώνει έναν τέλειο κατηγοριοποιητή, ενώ μια AUC με αριθμητική τιμή 0,5 αντιπροσωπεύει ένα μοντέλο που δεν είναι καλύτερο από την τυχαία κατηγοριοποίηση. Δεν υπάρχει ένας απλός τύπος για τον υπολογισμό της AUC. Υπολογίζεται γενικά χρησιμοποιώντας αριθμητικές μεθόδους, ή πιο πρακτικά, χρησιμοποιώντας μια συνάρτηση που παρέχεται από μια βιβλιοθήκη μηχανικής μάθησης.

(2.7.4 Απώλεια καταγραφής)

Η απώλεια καταγραφής, επίσης γνωστή ως λογαριθμική απώλεια ή λογιστική απώλεια, είναι μια ακόμη σημαντική μέτρηση για την αξιολόγηση μοντέλων κατηγοριοποίησης, ιδιαίτερα σε προβλήματα δυαδικής κατηγοριοποίησης. Σε αντίθεση με τις άλλες μετρήσεις που συζητήσαμε, η απώλεια καταγραφής λαμβάνει υπόψη τις βαθμολογίες πιθανοτήτων που παράγονται από τον κατηγοριοποιητή αντί για το τελικό αποτέλεσμα πρόβλεψης. Ένα από τα βασικά χαρακτηριστικά της απώλειας καταγραφής είναι ότι είναι ένας «αυστηρός» κανόνας βαθμολογίας. Αυτό σημαίνει ότι όχι μόνο απαιτεί τη

σωστή κατηγοριοποίηση, αλλά επίσης επιβραβεύει την εμπιστοσύνη στις σωστές κατηγοριοποιήσεις και τιμωρεί την εμπιστοσύνη σε λανθασμένες κατηγοριοποιήσεις. Ως εκ τούτου, παρέχει μια πιο λεπτή άποψη της απόδοσης ενός μοντέλου και μπορεί να είναι ιδιαίτερα χρήσιμη όταν μας ενδιαφέρει η αξιοπιστία των ίδιων των προβλέψεων και όχι μόνο εάν η τελική πρόβλεψη είναι σωστή ή λάθος. Η απώλεια καταγραφής υπολογίζεται χρησιμοποιώντας τις προβλεπόμενες πιθανότητες που δημιουργούνται από έναν κατηγοριοποιητή και τις αληθείς ετικέτες. Ορίζεται ως εξής για προβλήματα δυαδικής κατηγοριοποίησης:

$$\text{Απώλεια καταγραφής} = - \left(\frac{1}{n}\right) \sum y * \log(\hat{y}) + (1 - y) * \log(1 - \hat{y})$$

Όπου:

n είναι ο συνολικός αριθμός των παρατηρήσεων (δείγματα).

y είναι η πραγματική ετικέτα κλάσης του δείγματος (0 ή 1 σε περίπτωση δυαδικής κατηγοριοποίησης).

$\hat{y}$ είναι η προβλεπόμενη πιθανότητα ότι το δείγμα ανήκει στην κλάση 1.

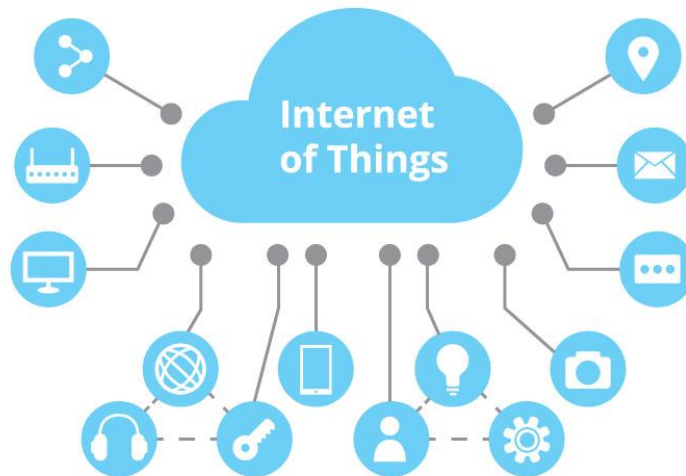
Η άθροιση Σ είναι πάνω από όλα τα δείγματα στο σύνολο δεδομένων.

Ουσιαστικά, αυτό που κάνει αυτός ο τύπος είναι να αθροίζει όλες τις μεμονωμένες απώλειες καταγραφής για κάθε δείγμα στο σύνολο δεδομένων και στη συνέχεια να παίρνει έναν μέσο όρο. Για κάθε μεμονωμένο δείγμα, η απώλεια καταγραφής υπολογίζεται λαμβάνοντας την καταγραφή της προβλεπόμενης πιθανότητας για την πραγματική κατηγορία στην οποία ανήκει το δείγμα. Εάν το δείγμα ανήκει στην κλάση 1 ($y=1$), η απώλεια καταγραφής είναι $-\log g(\hat{y})$ και αν το δείγμα ανήκει στην κλάση 0 ($y=0$), η απώλεια καταγραφής είναι $-\log(1 - \hat{y})$ [14].

ΚΕΦΑΛΑΙΟ 3 Διαχείριση δεδομένων στις παρυφές του δικτύου

(3.1 Internet of Things)

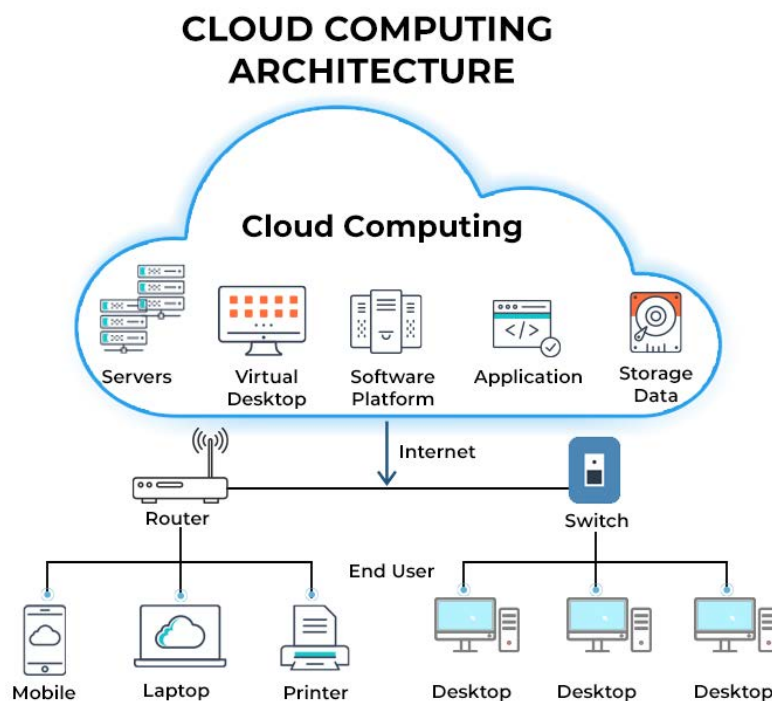
Η μελλοντική κατεύθυνση των υπολογιστών θα ξεπεράσει τον παραδοσιακό υπολογισμό ο οποίος είναι βασισμένος σε σταθερή επιφάνεια εργασίας. Ιδιαίτερα, το διαδίκτυο των πραγμάτων-IoT (Internet of Things) εντάσσεται στην καθημερινή μας ζωή γρήγορα, ως μια νέα τεχνολογία των τελευταίων ετών. Ως παράδειγμα, το IoT το οραματίζεται οι περισσότερες φυσικές συσκευές, όπως έξυπνα κινητά τηλέφωνα, οχήματα, αισθητήρες, ενεργοποιητές και οποιεσδήποτε άλλες ενσωματωμένες συσκευές να συνδέονται και να επικοινωνούν με κέντρα δεδομένων, να ανταλλάσσουν πληροφορίες ώστε πραγματοποιήσουν το επόμενο τεράστιο άλμα στην κλίμακα της παραγωγής δεδομένων. Ακολουθώντας διάφορες δημοφιλείς τεχνολογίες, όπως οι έξυπνες μεταφορές, η έξυπνη πόλη και έξυπνη υγειονομική περίθαλψη. Έτσι, το IoT θα επηρεάσει εντυπωσιακά την καθημερινή ζωή των χρηστών και θα είναι το κλειδί για το μέλλον. Το IoT επίσης διαδραματίζει σημαντικό ρόλο στον τομέα των επιχειρήσεων. Πράγματι, το IoT αναφέρθηκε ως μία από τις σημαντικότερες τεχνολογίες που θα επηρεάσει τα συμφέροντα των ΗΠΑ το 2025. Ομοίως, ο αριθμός των διασυνδεδεμένων φυσικών συσκευών θα έχει υπερβεί τον ανθρώπινο πληθυσμό ολόκληρου του κόσμου. Το 2012, ο αριθμός των διασυνδεδεμένων φυσικών συσκευών αυξήθηκε σε 9 δισεκατομμύρια. Οι συσκευές IoT θα είναι έτσι μία από τις πιο σημαντικές και επισκιάστηκες πηγές δεδομένων για τα Big data στο μέλλον [15].



Εικόνα 7: Internet of Things

(3.2 Cloud computing)

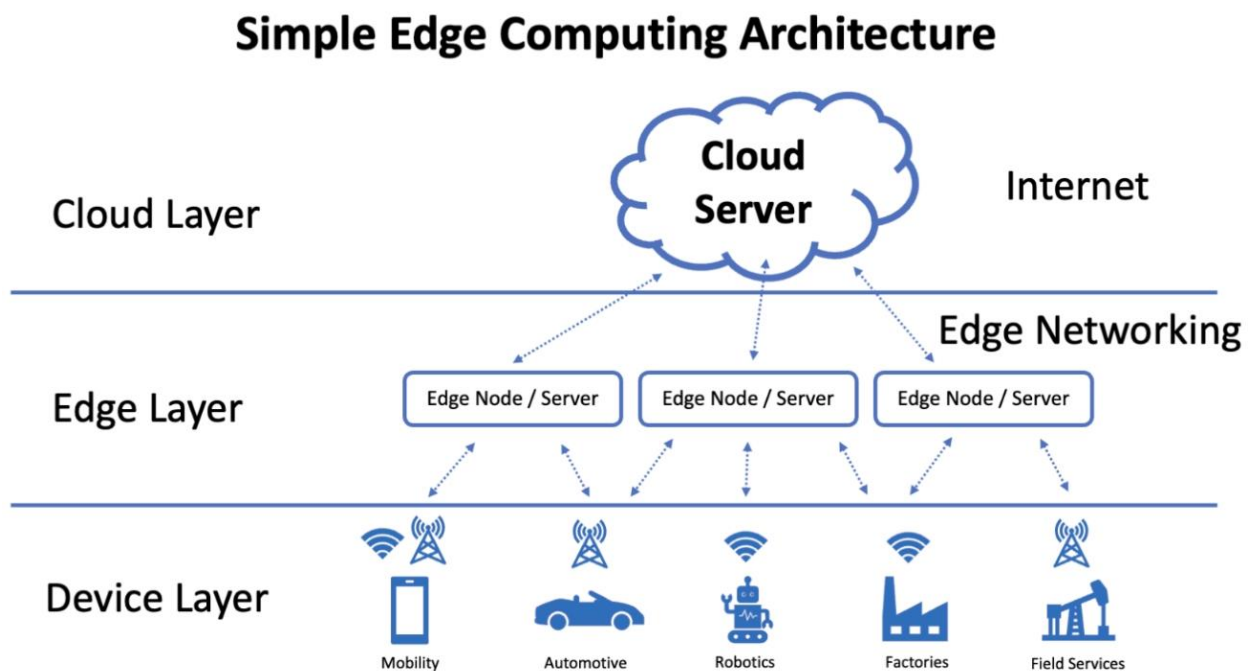
Το Cloud computing μεταφέρει όλα τα δεδομένα στο κέντρο του μέσω του δικτύου και επιλύει το υπολογιστικό και προβλήματα αποθήκευσης με κεντρικό τρόπο. Στη λογοτεχνία περιγράφεται το ιστορικό ανάπτυξης του Cloud computing. Στο συνέδριο της μηχανής αναζήτησης (sessane jose 2006) [18] τον Αύγουστο του 2006, ο Διευθύνων Σύμβουλος της Google πρότεινε για πρώτη φορά την ιδέα του Cloud computing. Στην ιστορία ανάπτυξης του Cloud computing, αυτή είναι η πρώτη φορά που υποβάλλεται επίσημα η έννοια του Cloud computing. Με την ανάπτυξη των μηχανών αναζήτησης που εκπροσωπούνται από την Google, ξεκινά το Cloud computing να δείξει έντονη ζωτικότητα. Στις μέρες μας, το Cloud computing έχει αναπτυχθεί σταδιακά. Είναι μια πολύ ισχυρή πλατφόρμα υπηρεσίας δικτύου συμπεριλαμβανομένων κατανεμημένων υπολογιστών, εξισορρόπηση φορτίων, παράλληλοι υπολογιστές, αποθήκευση δικτύου και άλλες τεχνολογίες. Ωστόσο, στις μέρες μας, με την ανάπτυξη του Διαδικτύου των Πραγμάτων στη ζωή των ανθρώπων, ο αριθμός των συσκευών που είναι συνδεδεμένες στο Internet of Things αυξάνεται σταδιακά και δημιουργείται μεγάλος όγκος δεδομένων. Το εύρος ζώνης δικτύου του υπολογιστικού νέφους ήταν ανίκανο να καλύψει τις ανάγκες των ευαίσθητων στο χρόνο συστημάτων και να έχει απόδοση σε πραγματικό χρόνο. Επομένως, μοντέλο υπολογιστικού νέφους έχει μεγάλα ελαττώματα στο φορτίο, σε πραγματικό χρόνο, στη μετάδοση του εύρους ζώνης, την κατανάλωση ενέργειας, την ασφάλεια και την προστασία του απόρρητου δεδομένων [18].



Εικόνα 8: Αρχιτεκτονική Υπολογιστικού νέφους

(3.3 Edge computing)

Τα δεδομένα παράγονται όλο και περισσότερο στην άκρη του δικτύου. Επομένως, θα ήταν πιο αποτελεσματικό να επεξεργάζονται τα δεδομένα στην άκρη του δικτύου. Ο υπολογισμός ακμών (Edge Computing) αναφέρεται στις τεχνολογίες που επιτρέπουν την εκτέλεση υπολογισμών στην άκρη του δικτύου, σε μεταγενέστερα δεδομένα για λογαριασμό των υπηρεσιών σύννεφου (cloud) και αναδρομικά δεδομένα για λογαριασμό των υπηρεσιών IoT. Εδώ ορίζουμε το "άκρο" ως οποιοδήποτε πόρο υπολογιστών και δικτύου κατά μήκος της διαδρομής μεταξύ των δεδομένων πηγών και των κέντρων δεδομένων του cloud. Για παράδειγμα, ένα έξυπνο τηλέφωνο είναι η άκρη μεταξύ των πραγμάτων του σώματος και του σύννεφου, μια πύλη σε ένα έξυπνο σπίνι είναι η άκρη μεταξύ οικιακών συσκευών και cloud, ένα κέντρο μικρών δεδομένων και ένα νέφος είναι η άκρη μεταξύ μιας κινητής συσκευής και του cloud. Η λογική του υπολογισμού ακμών είναι ότι ο υπολογισμός πρέπει να γίνεται κοντά σε πηγές δεδομένων. Από την άποψη μας, ο υπολογισμός ακμών είναι εναλλάξιμος με τον υπολογισμό ομίχλης, αλλά ο υπολογισμός ακμών εστιάζει περισσότερο στα πλευρικά πράγματα, ενώ ο υπολογισμός ομίχλης εστιάζει περισσότερο στην πλευρά υποδομής. Οραματιζόμαστε ότι το edge computing θα μπορούσε να έχει τόσο μεγάλο αντίκτυπο στην κοινωνία μας όσο και το cloud computing [16].



Εικόνα 9: Αρχιτεκτονική υπολογισμού ακμών

(3.4 Χαρακτηριστικά Edge computing)

Το Edge computing διαθέτει αρκετά χαρακτηριστικά παρόμοια με το Cloud computing. Ωστόσο, τα διακριτικά χαρακτηριστικά του Edge computing που το καθιστούν μοναδικό είναι τα εξής:

(3.4.1 Πυκνή γεωγραφική κατανομή)

Το Edge computing φέρνει τις υπηρεσίες Cloud πιο κοντά στον χρήστη αναπτύσσοντας πολυάριθμες πλατφόρμες υπολογιστών στα δίκτυα αιχμής. Η πυκνή γεωγραφική κατανομή της υποδομής βοηθά με τους ακόλουθους τρόπους: (α) Οι διαχειριστές του δικτύου μπορούν να διευκολύνουν τις υπηρεσίες κινητικότητας βάσει τοποθεσίας χωρίς να διασχίζουν ολόκληρο το WAN. (β) Η ανάλυση μεγάλων δεδομένων μπορεί να εκτελεστεί γρήγορα με καλύτερη ακρίβεια. (γ) Τα συστήματα ακμών επιτρέπουν την ανάλυση σε πραγματικό χρόνο σε μεγάλη κλίμακα. Παραδείγματα περιλαμβάνουν δίκτυα αισθητήρων για την παρακολούθηση του περιβάλλοντος και την παρακολούθηση αγωγών.

(3.4.2 Υποστήριξη κινητικότητας)

Καθώς ο αριθμός των κινητών συσκευών αυξάνεται ραγδαία, το Edge computing υποστηρίζει επίσης κινητικότητα, όπως το Πρωτόκολλο διαχωρισμού αναγνωριστικού εντοπισμού (LISP- Locator ID Separation Protocol), για άμεση επικοινωνία με κινητές συσκευές. Το πρωτόκολλο LISP αποσυνδέει την ταυτότητα τοποθεσίας από την ταυτότητα του κεντρικού υπολογιστή και υλοποιεί ένα κατακεντρωμένο σύστημα καταλόγου. Η αποσύνδεση της ταυτότητας του κεντρικού υπολογιστή από την ταυτότητα τοποθεσίας αποτελεί τη βασική αρχή που επιτρέπει την υποστήριξη κινητικότητας στον υπολογισμό ακμών.

(3.4.3 Επίγνωση τοποθεσίας)

Το χαρακτηριστικό εντοπισμού θέσης του Edge computing επιτρέπει στους χρήστες κινητών να έχουν πρόσβαση σε υπηρεσίες από αυτόν τον διακομιστή ακμών που βρίσκεται πιο κοντά στη φυσική τους τοποθεσία. Οι χρήστες μπορούν να χρησιμοποιήσουν διάφορες τεχνολογίες, όπως υποδομή κινητής τηλεφωνίας, GPS ή ασύρματα σημεία πρόσβασης για να βρουν τη θέση των ηλεκτρονικών συσκευών. Αυτή η επίγνωση θέσης μπορεί να χρησιμοποιηθεί από πολλές υπολογιστικές εφαρμογές ακμών.

(3.4.4 Εγγύτητα)

Στο Edge computing, οι υπολογιστικοί πόροι και οι υπηρεσίες είναι διαθέσιμες κοντά στους χρήστες που μπορούν να βελτιώσουν την εμπειρία τους. Η διαθεσιμότητα των υπολογιστικών πόρων και υπηρεσιών στην τοπική γειτνίαση επιτρέπει στους χρήστες

να αξιοποιήσουν τις πληροφορίες περιβάλλοντος δικτύου για τη λήψη αποφάσεις χρήσης της υπηρεσίας. Ομοίως, ο πάροχος υπηρεσιών μπορεί να αξιοποιήσει τις πληροφορίες του χρήστη του κινητού τηλεφώνου εξάγοντας πληροφορίες συσκευής και αναλύοντας τη συμπεριφορά του χρήστη προκειμένου να βελτιώσει τις υπηρεσίες του και την κατανομή πόρων.

(3.4.5 Χαμηλή καθυστέρηση)

Τα παραδείγματα Edge Computing φέρνουν τους υπολογιστικούς πόρους και τις υπηρεσίες πιο κοντά στους χρήστες, γεγονός που μειώνει τον λανθάνοντα χρόνο πρόσβασης στις υπηρεσίες. Η χαμηλή καθυστέρηση του Edge computing επιτρέπει στους χρήστες να εκτελούν τις εφαρμογές τους με ένταση πόρων (resource-intensive) και ευαίσθητες σε καθυστέρηση (delay-sensitive) στις πλούσιες σε πόρους συσκευές ακμών (π.χ. δρομολογητής, σημείο πρόσβασης, σταθμός βάσης ή αποκλειστικός διακομιστής).

(3.4.6 Επίγνωση του προβλήματος)

Η επίγνωση περιβάλλοντος είναι το χαρακτηριστικό των κινητών συσκευών που μπορεί να οριστεί αλληλεξαρτώμενα με την επίγνωση τοποθεσίας. Οι πληροφορίες περιβάλλοντος της κινητής συσκευής στο Edge computing μπορούν να χρησιμοποιηθούν για τη λήψη αποφάσεων εκφόρτωσης και την πρόσβαση στις υπηρεσίες Edge. Οι πληροφορίες δικτύου σε πραγματικό χρόνο, όπως το φόρτο δικτύου και η τοποθεσία του χρήστη, μπορούν να χρησιμοποιηθούν για την προσφορά υπηρεσιών με επίγνωση του περιβάλλοντος στους χρήστες των ακμών. Επιπλέον, ο πάροχος υπηρεσιών μπορεί να χρησιμοποιήσει τις πληροφορίες περιβάλλοντος για να βελτιώσει την ικανοποίηση των χρηστών και την ποιότητα της εμπειρίας.

(3.4.7 Ανομοιογένεια)

Η ετερογένεια στον υπολογισμό Edge αναφέρεται στην ύπαρξη ποικίλων πλατφορμών, αρχιτεκτονικών, υποδομών, υπολογιστών και τεχνολογιών επικοινωνίας που χρησιμοποιούνται από τα υπολογιστικά στοιχεία ακμών (τελικές συσκευές, διακομιστές Edge και δίκτυα). Στην ετερογένεια της τελικής συσκευής οι παραλλαγές λογισμικού, υλικού και τεχνολογίας αποτελούν τους κύριους παράγοντες της ετερογένειας. Η ετερογένεια από την πλευρά του διακομιστή Edge οφείλεται κυρίως σε API (Application Programming Interface), προσαρμοσμένες πολιτικές και πλατφόρμες. Τέτοιες υπάρχουσες διαφορές έχουν ως αποτέλεσμα ζητήματα διαλειτουργικότητας καθιστώντας την ως την κύρια πρόκληση στην επιτυχή ανάπτυξη του Edge computing. Η ετερογένεια δικτύου αναφέρεται στην ποικιλία των τεχνολογιών επικοινωνίας που επηρεάζουν την παροχή υπηρεσιών Edge [17].

(3.5 Πλεονεκτήματα Edge computing)

Το μοντέλο Edge computing αποθηκεύει και επεξεργάζεται δεδομένα στις συσκευές Edge χωρίς μεταφόρτωση σε πλατφόρμα Cloud computing.

	Applicable situation	Network bandwidth pressure	Real-time	Calculation mode
Cloud computing	Global	More	High	Large scale centralized processing
Edge computing	Local	Less	Low	Small scale intelligent analysis

Εικόνα 10: Κύριες διαφορές μεταξύ Cloud computing και Edge computing.

Λόγω αυτού του χαρακτηριστικού, το Edge computing έχει προφανή πλεονεκτήματα στις ακόλουθες πτυχές:

Γρήγορη επεξεργασία και ανάλυση δεδομένων, σε πραγματικό χρόνο: η γρήγορη αύξηση του όγκου δεδομένων και η πίεση του εύρους ζώνης δικτύου είναι μειονεκτήματα του Cloud computing. Σε σύγκριση το Cloud computing, το Edge computing έχει πλεονεκτήματα σε ταχύτητα απόκρισης και σε πραγματικό χρόνο. Το Edge computing είναι πιο κοντά στην πηγή δεδομένων, αποθήκευση δεδομένων και στις υπολογιστικές εργασίες που μπορούν να πραγματοποιηθούν στον κόμβο του Edge computing, ο οποίος μειώνει την ενδιάμεση διαδικασία μετάδοσης δεδομένων. Χαρακτηρίζεται από την εγγύτητα στους χρήστες και την παροχή έξυπνων υπηρεσιών στους χρήστες, βελτιώνοντας έτσι την απόδοση μετάδοσης δεδομένων, διασφαλίζοντας την επεξεργασία σε πραγματικό χρόνο και μειώνοντας το χρόνο καθυστέρησης. Edge computing παρέχει στους χρήστες μια ποικιλία υπηρεσιών γρήγορης απόκρισης, ειδικά στον τομέα της έξυπνης κατασκευής αυτόματης οδήγησης, της επίβλεψης βίντεο και άλλων εντοπισμών τοποθεσίας.

Ασφάλεια: Το Cloud computing απαιτεί όλα τα δεδομένα να μεταφορτωθούν στο Cloud για μαζική επεξεργασία, η οποία είναι μία κεντρική μέθοδος επεξεργασίας. Σε αυτή τη διαδικασία, θα υπάρξει κινδύνους όπως η απώλεια και η διαρροή δεδομένων, η διαδικασία αυτή δεν μπορεί να εγγυηθεί την ασφάλεια και το απόρρητο. Για παράδειγμα, κωδικοί πρόσβασης λογαριασμού, ιστορικά αρχεία αναζήτησης και ακόμη και εμπορικά μυστικά μπορούν όλα να εκτεθούν. Δεδομένου ότι το Edge computing είναι υπεύθυνο μόνο για την τοπική επεξεργασία των δεδομένων, δεν χρειάζεται να ανεβούν στο Cloud, για να αποφευχθούν οι κίνδυνοι που επιφέρει η διαδικασία μετάδοσης του δικτύου, ώστε η ασφάλεια των δεδομένων να μπορεί να είναι εγγυημένη. Όταν τα δεδομένα δέχονται επίθεση, επηρεάζονται μόνο τα τοπικά δεδομένα.

Χαμηλό κόστος, χαμηλή κατανάλωση ενέργειας, χαμηλό κόστος εύρους ζώνης: στο Edge computing, τα προς επεξεργασία δεδομένα δεν χρειάζεται να μεταφορτωθούν στο κέντρο του Cloud computing, εφόσον δεν υπάρχει η ανάγκη για να χρησιμοποιηθεί πολύ μεγάλο εύρος ζώνης δικτύου, το φορτίο του εύρους ζώνης του δικτύου μειώνεται και η κατανάλωση ενέργειας ευφυών συσκευών στην ακμή του δικτύου είναι πολύ μειωμένη. Το Edge computing είναι «μικρής κλίμακας» και στην παραγωγή, οι εταιρείες μπορούν να μειώσουν το κόστος επεξεργασίας δεδομένων με ιδιόκτητο εξοπλισμό. Επομένως, το Edge computing μειώνει την ποσότητα των δεδομένων που μεταδίδονται στο δίκτυο, μειώνει το κόστος μετάδοσης και την πίεση εύρους ζώνης του δικτύου, μειώνει την κατανάλωση ενέργειας του ιδιόκτητου εξοπλισμού και βελτιώνει την υπολογιστική αποδοτικότητα [18].

(3.6 Εφαρμογές του IoT που βασίζονται σε Edge)



Εικόνα 11: Εφαρμογές υπολογισμού ακμών

1) Έξυπνα συστήματα κυκλοφορίας: Σε δίκτυα οχημάτων που αναπτύσσονται. Αρχιτεκτονικές MEC (Multi-Access Edge Computing) ή ομίχλης, οχήματα και ακίνητοι διακομιστές Edge σχηματίζουν δίκτυα ad-hoc για την προσωρινή αποθήκευση τοπικού περιεχομένου και κοινή χρήση δεδομένων αισθητήρα. Τα οχήματα παράγουν συνεργατικά σε πραγματικό χρόνο χάρτες του κυκλοφοριακού συστήματος. Τα αυτοκίνητα που εισέρχονται στην περιοχή αυτή στη συνέχεια συλλέγουν το περιεχόμενο από διακομιστές Edge. Το Cloud παρέχει πληροφορίες που λείπουν και αναλύει δεδομένα για βελτιστοποίηση συστήματος. Τέτοιες προσεγγίσεις επιτρέπουν συστήματα πρόληψης κινδύνων, όπου οι διακομιστές Edge εκδίδουν προειδοποιήσεις σε πραγματικό χρόνο σε όλα τα αυτοκίνητα στην περιοχή ενός περιστατικού. Η Bonomi et al [19], περαιτέρω προτείνει ένα έξυπνο σύστημα φωτεινών σηματοδοτών, όπου οι

κόμβοι ομίχλης (Fog) αναλύουν δεδομένα αισθητήρων τόσο των οχημάτων που πλησιάζουν όσο και των πεζών που διασχίζουν το δρόμο. Με βάση τα αποτελέσματα, στέλνουν σε πραγματικό χρόνο προειδοποιήσεις στα οχήματα και προσαρμόζουν τους κύκλους των φωτεινών σηματοδοτών.

2) Έξυπνα δίκτυα: Έξυπνες συσκευές μέτρησης ενσωματωμένες στο ηλεκτρικό δίκτυο συγκεντρώνουν τα δεδομένα σε πραγματικό χρόνο για την κατανάλωση ενέργειας οντοτήτων (π.χ. Έξυπνα σπίτια) που συνδέονται με αυτά για ανάλυση της κατανάλωσης ενέργειας των νοικοκυριών. Εφόσον η ανάλυση Cloud εξαρτάται από τη συνδεσιμότητα στο Διαδίκτυο και καταναλώνει μεγάλο εύρος ζώνης, μια ιεραρχική πλατφόρμα ομίχλης έχει προταθεί, διασυνδέοντας έξυπνους μετρητές στην ακμή της διεργασίας δικτύου με ευαισθησία σε καθυστέρηση μεταφοράς τοπικά δεδομένα και εκδίδοντας εντολές στους ενεργοποιητές σε πραγματικό χρόνο για διόρθωση ανωμαλιών. Τα δεδομένα με ανοχή σε καθυστέρηση μεταφοράς μεταβιβάζονται σε υψηλότερα επίπεδα για ανάλυση και αναφορά.

3) Έξυπνο σπίτι: Προσαρτημένοι ασύρματοι αισθητήρες και ενεργοποιητές στις συσκευές ελέγχουν τις συνθήκες περιβάλλοντος και επιτρέπουν τη διαχείριση οικιακών συσκευών. Από την επικοινωνία μεταξύ συσκευών και εξωτερικών οντοτήτων (π.χ. Έξυπνα δίκτυα) δημιουργείται συμφόρηση δικτύου, μπορούν να αναπτυχθούν διακομιστές ομίχλης για να εντοπιστούν ανωμαλίες στα δεδομένα των αισθητήρων και να δώσουν οδηγίες στους ενεργοποιητές ώστε να προσαρμοστούν στις συνθήκες δόμησης (π.χ. ανοίξτε ένα παράθυρο) ή να στείλουν αιτήματα συντήρησης στην συμβαλλόμενη εταιρεία. Η Li et al [19], αποδεικνύει ότι η καθυστέρηση μπορεί να μειωθεί κατά 73%. Επιπλέον, ο υπολογισμός ομίχλης προστατεύει το απόρρητο με την επεξεργασία δεδομένων εντός του τοπικού δικτύου, και μπορεί να ενισχύσει την ασφάλεια του κτιρίου, οι κάμερες παρακολούθησης, οι αισθητήρες και οι ενεργοποιητές συνεργάζονται για την ανίχνευση κινδύνων και εισβολών. Κάθε αισθητήρας συνδέεται με έναν κόμβο ομίχλης που λειτουργεί μια εφαρμογή ασφαλείας. Σε περίπτωση ανίχνευσης κινδύνου (π.χ. ύποπτη κίνηση), μπορεί να ζητήσει ένας διακομιστής ομίχλης υψηλότερου επιπέδου πρόσθετα δεδομένα αισθητήρα για επιβεβαίωση. Ο κόμβος ομίχλης τότε είτε ειδοποιεί τον ιδιοκτήτη είτε τις εξωτερικές αρχές.

4) Έξυπνη περίθαλψη: Δεδομένα από δίκτυα αισθητήρων σώματος επιτρέπουν προηγμένες ιατρικές υπηρεσίες, αλλά η επεξεργασία στο Cloud επιβαρύνει το δίκτυο και οδηγεί σε υψηλές καθυστερήσεις. Ως εκ τούτου, οι συγγραφείς στο [19] αναλύουν τα δεδομένα του αισθητήρα ΗΚΓ (ECG-Electrocardiogram) για κόμβους ομίχλης για την ανίχνευση μη φυσιολογικών καρδιακών παλμών. Τα δεδομένα προωθούνται σε πραγματικό χρόνο έγκαιρα στον γιατρό για να σωθεί ο ασθενής. Τα επαναλαμβανόμενα δεδομένα συγκεντρώνονται πριν σταλούν στο Cloud για διάγνωση και παρακολούθηση, μειώνοντας τα μεταδιδόμενα δεδομένα κατά 90% και εξοικονομώντας ενέργεια στους κόμβους αισθητήρων.

5) Βιομηχανικός Αυτοματισμός: Η επικοινωνία M2M(Machine to Machine) ενεργοποιεί βιομηχανικές μηχανές για την ολοκλήρωση των εργασιών κατασκευής με ελάχιστη ανθρώπινη παρέμβαση. Οι διακομιστές ομίχλης εκτελούν αλγορίθμους μηχανικής μάθησης για την κατανόηση της κανονικής συμπεριφοράς της συσκευής και στέλνουν δεδομένα στον χειριστή μόνο εάν συμβαίνει κάποια ανωμαλία. Οι Stojmenovic et al [19] προτείνει δύο περιπτώσεις χρήσης: Στην πρώτη περίπτωση, οι αισθητήρες που αναπτύσσονται στα τρένα ανιχνεύουν τα επίπεδα θερμότητας, επιτρέποντας την αυτόματη λειτουργία εφαρμογών που εκτελούνται σε διακομιστές ομίχλης ειδοποιώντας τον οδηγό για οποιαδήποτε συντήρηση έκτακτης ανάγκης. Στη δεύτερη περίπτωση, αισθητήρες στα ορυχεία παρακολουθούν τις συνθήκες του αέρα και ρυθμίζουν τη ροή του αέρα.

6) Έξυπνη πόλη: Έξυπνες πόλεις συνδυάζουν πολλούς τομείς IoT, βελτιώνοντας την εμπειρία χρήστη, συστήματα δια δραστικών οδηγών παρουσιάζουν οι Taleb et al [19] έναν τουριστικό οδηγό, όπου οι επισκέπτες μεταδίδουν πληροφορίες πολυμέσων σε αξιοθέατα από διακομιστές Edge. Καθώς περιφέρονται, το περιεχόμενο είναι μεταφέρεται, ενεργοποιώντας τη ροή βίντεο HD. Εάν προσφέρονται οδηγοί όπως οι εφαρμογές επαυξημένης πραγματικότητας (AR), οι διακομιστές Edge παρέχουν επεξεργασία χαμηλής καθυστέρησης υπολογίζοντας την θέση του χρήστη καλύπτοντας απευθείας τοπικό περιεχόμενο [19].

ΚΕΦΑΛΑΙΟ 4 Προτεινόμενο μοντέλο

Σε αυτό το κεφάλαιο παρουσιάζουμε το μοντέλο που αναπτύχθηκε, το οποίο έχει σχεδιαστεί για την διαχείριση και βελτιστοποίηση των ροών ερωτημάτων σε ένα δίκτυο κόμβων ομαδοποιώντας, συγχωνεύοντας και συγκρίνοντας ερωτήματα και δεδομένα μεταξύ κόμβων. Αρχικά, με την χρήση συστάδων ομαδοποιούνται παρόμοια ερωτήματα σε κάθε κόμβο με βάση τις επικαλυπτόμενες περιοχές, συγχωνεύοντας ύστερα τις παρόμοιες συστάδες για να πραγματοποιηθεί περαιτέρω επεξεργασία. Οι περισσότερες από τις μεθόδους συσταδοποίησης ροών πρέπει να γνωρίζουν εκ των προτέρων πόσες συστάδες μπορούν να βρεθούν σε μια ροή δεδομένων ή να εξαρτώνται τουλάχιστον από μία διαφορετική παραμετροποίηση για το αποτέλεσμα τους. Ωστόσο, αντιμετωπίζοντας δυναμικά περιβάλλοντα όπου η κατανομή των ροών δεδομένων μπορεί να αλλάξει με την πάροδο του χρόνου. Δημιουργείται η ανάγκη για προσαρμοστικούς αλγόριθμους συστάδων που προσαρμόζουν τις παραμέτρους και τον τρόπο συσταδοποίησης των δεδομένων με βάση τις αλλαγές των ροών δεδομένων [20]. Για την δημιουργία των ερωτημάτων, πραγματοποιήσαμε συγχώνευση των παραγόμενων συστάδων συγχωνεύοντας τα σχετικά ερωτήματα σε μεγαλύτερες συστάδες. Με αυτήν την μέθοδο αποφεύγουμε τις ανακρίβειες (όπως είναι τα περιττά ερωτήματα ή τους υπερφορτωμένους κόμβους). Μέσω της συσταδοποίησης και της συγχώνευσης, το σύστημα βελτιστοποιεί την διαχείριση ερωτημάτων, διασφαλίζοντας ότι παρόμοιες εργασίες ομαδοποιούνται πριν γίνουν προβληματικές. Μέσα από τη απομάκρυνση των θορυβώδη δεδομένων, το σύστημα διατηρεί υψηλή απόδοση εστιάζοντας στα πιο σημαντικά ερωτήματα, διασφαλίζοντας την μελλοντική ομαλή διαχείριση του φόρτου εργασίας. Τέλος, συγκρίνουμε αν οι κόμβοι που περιέχουν τα δεδομένα συμπίπτουν με τους κόμβους που περιέχουν τα ερωτήματα, ώστε να δούμε τι ποσοστό των ερωτημάτων απαντάται.

(4.1 Gaussian distribution)

Gaussian κατανομή: Η Gaussian (ή κανονική) κατανομή είναι μια συνεχής κατανομή πιθανότητας, η οποία είναι η πιο σημαντική κατανομή και πιο δημοφιλής επειδή θεωρείται ότι η τα γεγονότα που παρατηρούνται στην καθημερινή μας ζωή αντιστοιχούν συνήθως σε αυτή την κατανομή. Η χρήση της κατανομής αυτής στην εργασία αυτή αποσκοπεί στην δημιουργία των ερωτημάτων και των δεδομένων. Η κατανομή Gauss είναι πολύ σημαντική και χρησιμοποιείται συχνά στη φύση και την κοινωνική επιστήμη για να αναπαραστήσει πραγματικές τιμές τυχαίων μεταβλητών. Υπάρχουν πολλά παραδείγματα τυχαίων μεταβλητών που ακολουθούν αυτήν την κατανομή, όπως οι μηνιαίες αποδόσεις μιας επένδυσης, το βάρος των παραγόμενων προϊόντων και τα αποτελέσματα των τεστ IQ. Χρησιμοποιείται επίσης ως βασική κατανομή σε στατιστικά συμπεράσματα. Ο γενικός τύπος για τη συνάρτηση πυκνότητας πιθανότητας έχει ως εξής :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

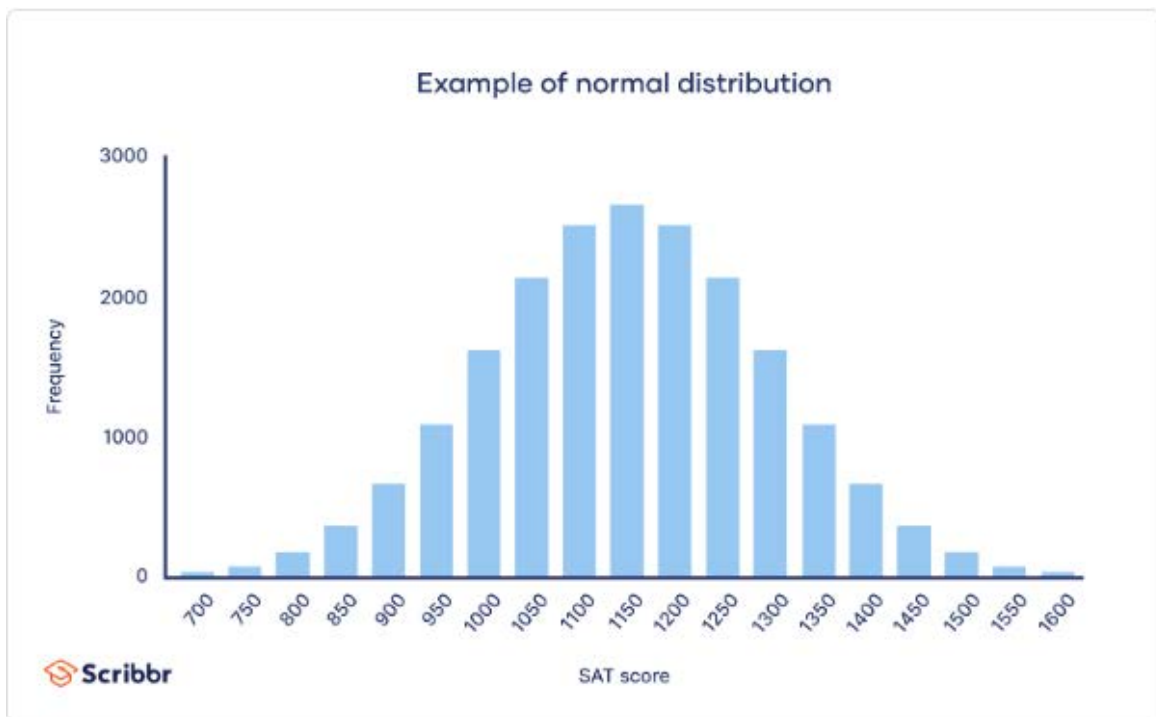
όπου x είναι μια τυχαία μεταβλητή, το μ και το σ^2 είναι ο μέσος όρος και οι παράμετροι διακύμανσης της κατανομής, και σ είναι η τυπική απόκλιση. Ορίζεται ο μέσος όρος ($\bar{x}$) και η τυπική απόκλιση (σ) ενός δείγματος

ως εξής:

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

όπου n είναι ο αριθμός των στοιχείων στο δείγμα [21].



Εικόνα 12: Παράδειγμα κανονικής κατανομής

(4.2 Αφαίρεση ακραίων τιμών με την χρήση του Z-score)

Το τυπικό Z-Score υπολογίζει την απόσταση μεταξύ των σημείων δεδομένων και τον μέσο όρο διαιρούμενο με την τυπική απόκλιση. Το Z-Score μπορεί να είναι θετικό ή αρνητικό. Εάν η τιμή z παραμένει πάνω από τη μέση τιμή, είναι θετική, διαφορετικά είναι αρνητική. Ο τύπος για τον υπολογισμό του Z-Score είναι ο εξής:

$$Z = \frac{x - \mu}{\sigma}$$

Όπου, η μεταβλητή x είναι τα δεδομένα μας, μ είναι η μέση τιμή και σ είναι η τυπική απόκλιση. Ο αλγόριθμος θα προσδιορίσει τις ακραίες τιμές χρησιμοποιώντας το Z-Score, δείχνοντας εάν ένα σημείο δεδομένων απέχει πολύ από τον μέσο όρο του πληθυσμού [22].

(4.3 Αλγόριθμοι συσταδοποίησης που χρησιμοποιήθηκαν)

Το IoT χρειάζεται την μηχανική μάθηση για τη συλλογή και ανάλυση δεδομένων για τη λήψη έξυπνων αποφάσεων. Ο σχεδιασμός, η ανάλυση και η πρόβλεψη είναι τα τρία κύρια καθήκοντα και όλα περιλαμβάνουν μηχανική μάθηση [23]. Σε αυτήν την εργασία τα ερωτήματα τα διαχειριζόμαστε μέσω μιας απλής συνδεδεμένης λίστας τα οποία υπάρχουν μέσα σε κάθε κόμβο σε μορφή $\{\text{[min, max]}, [\text{min, max}], [\text{min, max}], \dots\}$, κάθε κόμβος είναι διαιρεμένος σε διαστάσεις. Εδώ έρχονται σε χρήση οι αλγόριθμοι συσταδοποίησης οι οποίοι θα χρησιμοποιηθούν για την ομαδοποίηση των ερωτημάτων σε κάθε διάσταση.

Η συσταδοποίηση είναι μια ευρέως διαδεδομένη μεθοδολογία μηχανικής μάθησης η οποία δεν απαιτεί καμία διαδικασία εκπαίδευσης και παρέχει το τελικό αποτέλεσμα συσταδοποίησης σε περιορισμένο χρονικό διάστημα. Πρέπει να σημειωθεί ότι ο αριθμός των συστάδων είναι προκαθορισμένος και απεικονίζει την καλύτερη δυνατή τιμή για την επίτευξη συστάδων υψηλής ποιότητας. Ωστόσο, μπορούμε επίσης να βασιστούμε σε μεθόδους συσταδοποίησης που δεν απαιτούν τον αριθμό των συστάδων εκ των προτέρων αυξάνοντας την αυτόνομη φύση του προτεινόμενου πλάνου. Η διαδικασία συσταδοποίησης λαμβάνει ένα σύνολο συστάδων

$$C = \{C_1^J, C_2^J, \dots, C_{|C|}^J\}_{j=1}^M$$

$$\text{με κεντροειδή : } \{c_1^J, c_2^J, \dots, c_{|C|}^J\}_{j=1}^M$$

έτσι ώστε να ισχύει η ακόλουθη εξίσωση:

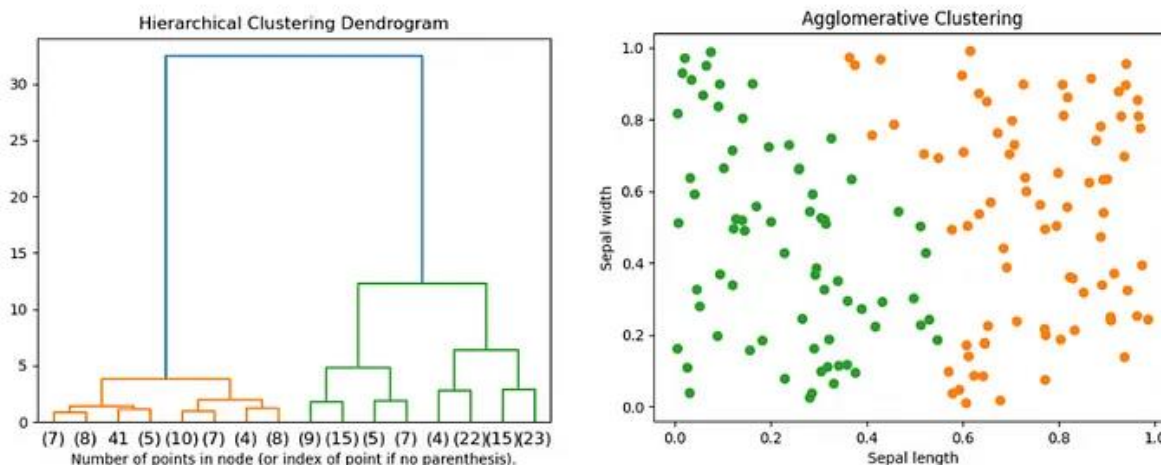
$$c_m^J = \frac{1}{|C_m^J|} \sum_{i=1}^{|C_m^J|} \|\tilde{y}_{ij}^J, \tilde{y}_{ij}^h\| - \|\tilde{y}_{i'j}^J, \tilde{y}_{i'j}^h\|, \forall (i, i'), i \neq i' \quad [24]$$

Το 2005, προτάθηκε μια προσέγγιση μη επιβλεπόμενης μάθησης βασισμένη στην πυκνότητα και στην συσταδοποίηση βάση πλέγματος για την ακριβή κατηγοριοποίηση

του συνόλου δεδομένων υψηλών διαστάσεων σε ένα σύνολο συστάδων, εκείνα τα σημεία που δεν συμπίπτουν σε κανένα σύμπλεγμα σημειώνονται ως μη φυσιολογικά [23].

(4.3.1 Αθροιστική συσταδοποίηση-Agglomerative Clustering)

Η ιεραρχική συσταδοποίηση μπορεί να γίνει με δύο τρόπους, συγκεντρώνοντας από μεμονωμένα σημεία στη συστάδα του πιο υψηλού επιπέδου ή διαιρώντας από μία κορυφαία συστάδα σε αντικείμενα ατομικών δεδομένων. Η εστίασή μας είναι η προσέγγιση από κάτω προς τα πάνω, η οποία είναι γνωστή ως αθροιστική (agglomerative) προσέγγιση, επειδή το υπολογιστικό κόστος μπορεί να μειωθεί εάν η διαδικασία από κάτω προς τα πάνω ξεκινήσει από κάπου στη μέση της ιεραρχίας και το κάτω μέρος της ιεραρχίας χτιστεί από μία λιγότερο ακριβή μέθοδο όπως η τμηματική συσταδοποίηση. Αυτή η ιδέα δεν θα λειτουργούσε καλά στην προσέγγιση από πάνω προς τα κάτω, η οποία είναι γνωστή ως διαιρετική (divisive) ιεραρχική συσταδοποίηση, επειδή είναι διαβόητη για το υψηλό κόστος της, $O(2^n)$ και η επαλήθευση υπο-συστάδων μεσαίου επιπέδου ανά μεμονωμένα σημεία δεδομένων θα εξακολουθούσε να είναι δαπανηρή [25].



Εικόνα 13: Agglomerative clustering

(4.3.2 DBSCAN- Χωρική Ομαδοποίηση Εφαρμογών με Θόρυβο με Βάση την Πυκνότητα)

DBSCAN (Density Based Spatial Clustering of Applications with Noise) είναι ένας βασικός αλγόριθμος πυκνότητας βασισμένος συσταδοποίηση. Απαιτεί δύο καθολικές εισαγωγές παραμέτρων που καθορίζονται από το χρήστη, τον ελάχιστο αριθμό αντικειμένων (MinPts) και την ακτίνα (Eps). Η πυκνότητα ενός αντικειμένου είναι ο αριθμός των αντικειμένων στη Eps-γειτονιά αυτού του αντικειμένου. Ο DBSCAN δεν καθορίζει το ανώτερο όριο ενός αντικειμένου πυρήνα, δηλαδή πόσα αντικείμενα μπορεί να υπάρχουν στη Eps-γειτονιά του. Έτσι, λόγω αυτού, οι συστάδες που ανιχνεύονται

από αυτό, έχουν ευρεία διακύμανση στην τοπική πυκνότητα. Τέτοιες συστάδες μπορεί να αντιπροσωπεύονται από αρκετά μικρότερες συστάδες έτσι ώστε κάθε συστάδα να μπορεί να έχει ομοιόμορφη πυκνότητα.

Ορ.1: Η Eps-γειτονιά ενός αντικειμένου p , που συμβολίζεται με $N(p)$, ορίζεται ως ο συνολικός αριθμός αντικειμένων που βρίσκονται στην ακτίνα, δηλ.

$$NEps(p) = \{q \in D \mid \text{dist}(p, q) \leq Eps\}$$

Ορ.2: Ένα αντικείμενο p λέγεται ότι είναι αντικείμενο πυρήνα εάν

$$|NEps(p)| \geq \text{MinPts (ελάχιστα αντικείμενα)}.$$

Ορ.3: Ένα αντικείμενο p λέγεται ότι είναι άμεσα χωρικά προσβάσιμο από ένα αντικείμενο q σε σχέση με Eps και MinPts αν το $p \in NEps(q)$ και το q είναι αντικείμενο πυρήνα.

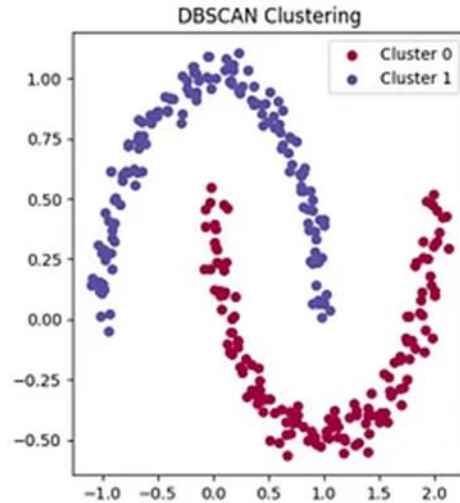
Ορ.4: Ένα αντικείμενο p λέγεται ότι είναι χωρικά προσβάσιμο από ένα αντικείμενο q αν υπάρχει μια αλυσίδα αντικειμένων $p_1, \dots, p_n$, $p_1 = q$, $p_n = p$ έτσι ώστε $p_i + 1$ είναι άμεσα χωρικά προσβάσιμη από το p_i ως προς Eps και MinPts.

Ορ.5: Ένα αντικείμενο p λέγεται ότι είναι χωρικά συνδεδεμένο με ένα αντικείμενο q σε σχέση με Eps και MinPts αν υπάρχει ένα αντικείμενο o έτσι ώστε και τα δύο, p και q είναι χωρικά προσβάσιμα από το o ως προς Eps και MinPts.

Ορ.6: Ένα αντικείμενο που βρίσκεται στα όρια δεν είναι αντικείμενο πυρήνα, αλλά θα είναι μέρος της συστάδας. Ένα αντικείμενο που δεν είναι που βρίσκεται σε οποιοδήποτε από τις συστάδες αντιμετωπίζεται ως αντικείμενο θορύβου.

Ορ.7: Μία συστάδα X δεν είναι κενό υποσύνολο της βάσης δεδομένων ως προς το Eps έως και MinPts, για κάθε p, q : αν $p \in X$, q είναι χωρικά προσβάσιμο από το p τότε το $q \in X$ και το p είναι χωρικά συνδεδεμένο με το q .

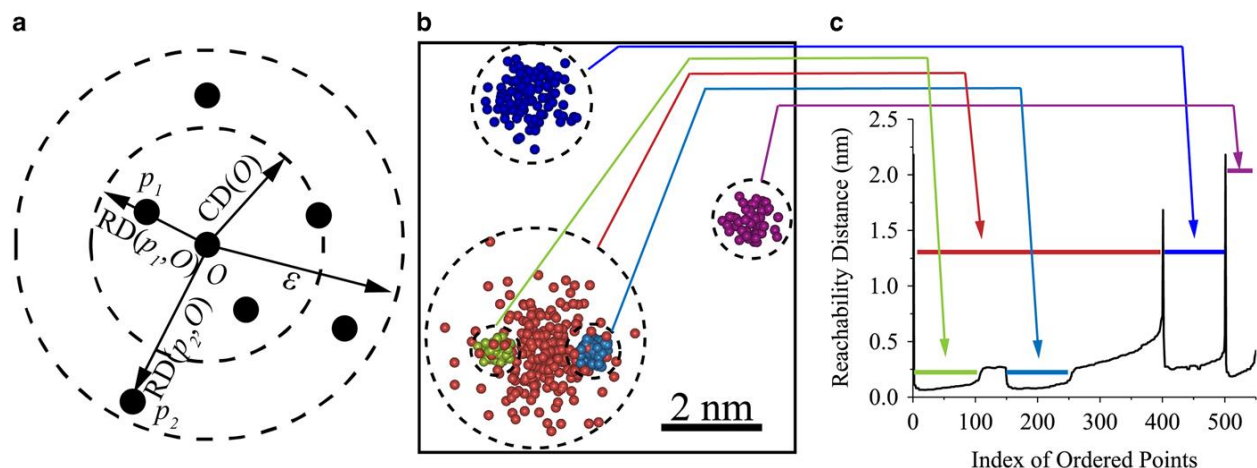
Ο DBSCAN ανιχνεύει χωρικά συνδεδεμένες συστάδες ανακαλύπτοντας ένα από τα στοιχεία του πυρήνα του p και υπολογίζει όλα τα αντικείμενα που είναι χωρικά προσβάσιμα από p . Η συλλογή των χωρικά προσβάσιμων αντικειμένων εκτελείται με επαναληπτικό υπολογισμό άμεσα χωρικών πρόσβασης αντικειμένων. Ο DBSCAN ελέγχει την Eps-γειτονιά του καθενός αντικειμένου p στη βάση δεδομένων. Αν το $NEps(p)$ ενός αντικειμένου p αποτελείται από τουλάχιστον MinPts αντικείμενα, δηλαδή, εάν το p είναι ένα αντικείμενο πυρήνα, δημιουργείται μία νέα συστάδα X που περιέχει όλα τα αντικείμενα του $NEps(p)$. Στη συνέχεια, η Eps-γειτονιά όλων των αντικειμένων $q \in X$, που δεν έχουν γίνει ακόμη επεξεργαστεί, ελέγχονται. Αν το αντικείμενο q είναι επίσης αντικείμενο πυρήνα, οι γείτονες του q , που δεν έχουν ήδη εκχωρηθεί στη συστάδα X , προστίθενται στο X και η Eps-γειτονιά τους ελέγχεται στο επόμενο βήμα. Αυτή η διαδικασία επαναλαμβάνεται έως ότου δεν μπορεί να προστεθεί νέο αντικείμενο στην τρέχουσα συστάδα X [26].



Εικόνα 14 : DBSCAN CLUSTERING

(4.3.3 OPTICS)

Όταν ο DBSCAN χρησιμοποίησε μια δυαδική δήλωση πυκνότητας, ο OPTICS(Ordering Points To Identify the Clustering Structure) διατηρεί μόνο τον ελάχιστο αριθμό αντικείμενων MinPts. Στον OPTICS, ένα σημείο είναι πυκνό σε μια ακτίνα r εάν έχει τουλάχιστον MinPts-γείτονες εντός αυτής της ακτίνας. Ο OPTICS είναι ένας απλός αλγόριθμος, ο οποίος επεξεργάζεται πάντα πρώτα τους γείτονες των πιο πυκνών σημείων (οργανώνοντας τα σημεία σε μια ουρά προτεραιότητας) και παράγει μια σειρά των σημείων που είναι γνωστή ως "σειρά συστάδων". Το διάγραμμα τοποθετεί αντικείμενα στον άξονα x σύμφωνα με τη σειρά συστάδων τους και την πυκνότητα στην οποία έφτασε το σημείο στον άξονα y , είναι γνωστό ως «γραφική παράσταση OPTICS». Εάν μια συστάδα εμφανίζει δομή συσταδοποίησης πυκνότητας, αυτή η γραφική παράσταση θα εμφανίσει κοιλάδες που αντιστοιχούν στις συστάδες. Εάν υπάρχει μια ιεραρχία συστάδων, μπορεί να υπάρχουν μικρότερες κοιλάδες μέσα σε μεγαλύτερες κοιλάδες [27].



Εικόνα 15 : OPTICS Clustering

(4.4 Συνδυασμός και διαχείριση επικαλυπτόμενων συστάδων)

Έπειτα από την χρήση των παραπάνω αλγορίθμων συστάδων για κάθε διάσταση του κάθε κόμβου οι συστάδες οι οποίες συμπίπτουν μεταξύ τους (τα όρια τα οποία ορίζονται από τις ελάχιστες/μέγιστες τιμές) συγχωνεύονται σε μια συστάδα. Για κάθε συστάδα, ελέγχουμε αν υπάρχει επικάλυψη με κάποια από τις προηγούμενες επεξεργασμένες συστάδες. Άμα βρεθεί επικάλυψη, τα όρια της συστάδας συγχωνεύονται αυξάνοντας τις ελάχιστες και μέγιστες τιμές της. Αντιθέτως αν δεν βρεθεί επικάλυψη, η συστάδα θα παραμείνει αυτούσια. Οι συγχωνευμένες συστάδες αποθηκεύονται στο ``combined_clusters`` για περαιτέρω επεξεργασία.

(4.5 Υπολογισμός ποσοστού επιτυχίας μεταξύ κόμβων)

Για τον υπολογισμό του ποσοστού επιτυχίας πραγματοποιήθηκε ανάλυση μεταξύ των κόμβων που περιέχουν τα ερωτήματα και των κόμβων που περιέχουν τα δεδομένα. Για την αξιολόγηση του ποσοστού επιτυχίας, έγινε σύγκριση των δεδομένων κάθε κόμβου με τα όρια των κόμβων των ερωτημάτων, ελέγχοντας εάν τα σημεία δεδομένων συμπίπτουν στα όρια αυτά, υπολογίζοντας το ποσοστό των δεδομένων που απαντούσε στα ερωτήματα. Στην συνέχεια, πραγματοποιήθηκε η εξαγωγή των ποσοστών επιτυχίας για κάθε σύγκριση, τα οποία συγκεντρώθηκαν για να αξιολογηθεί το συνολικό ποσοστό επιτυχίας. Το αποτέλεσμα αυτής της μεθόδου είναι η αξιολόγηση της ικανότητας των δεδομένων να απαντήσουν στις ροές των ερωτημάτων.

ΚΕΦΑΛΑΙΟ 5 Πειραματική αποτίμηση

Καθώς οι σύγχρονες εφαρμογές που βασίζονται σε δεδομένα εξελίσσονται συνεχώς, η ανάγκη για την επεξεργασία και ανάλυση δεδομένων σε πραγματικό χρόνο γίνεται ολοένα και σημαντικότερη. Ο τεράστιος όγκος δεδομένων που παράγονται από διάφορες πηγές όπως είναι οι συσκευές IoT, οι αισθητήρες και οι αλληλεπιδράσεις μεταξύ χρηστών απαιτούν συστήματα τα οποία μπορούν να επεξεργάζονται, αναλύουν και ανταποκρίνονται γρήγορα στις μεταβαλλόμενες συνθήκες. Το EC παρουσιάζεται ως μια πολλά υποσχόμενη λύση, φέρνοντας τον υπολογισμό πιο κοντά στις πηγές δεδομένων, μειώνοντας την εξάρτηση από κεντρικούς διακομιστές cloud και ενεργοποιώντας συστήματα με μεγαλύτερη απόκριση. Στο παρόν κεφάλαιο, θα διερευνήσουμε την ενσωμάτωση του EC με την ευφυή προδραστική αναγνώριση μέσα από ροές ερωτημάτων, χρησιμοποιώντας μια προσέγγιση που βασίζεται σε αλγορίθμους συσταδοποίησης για την ανάλυση και επεξεργασία δεδομένων των ακμών. Το προτεινόμενο σύστημα, το οποίο αποτυπώνεται μέσω του κώδικα που υλοποιήθηκε, χρησιμοποιεί διάφορες τεχνικές συσταδοποίησης, όπως είναι ο DBSCAN, ο OPTICS και το Agglomerative Clustering. Με σκοπό την πραγματοποίηση αυτοματοποιημένης ανίχνευσης ομάδων παρόμοιων σημείων δεδομένων χωρίς να απαιτείται προηγούμενη γνώση του αριθμού συστάδων, τον χειρισμό θορυβώδη δεδομένων αναγνωρίζοντας ακραίες τιμές που βρίσκονται εκτός των καθορισμένων ορίων των συστάδων και τον συνδυασμό επικαλυπτόμενων συστάδων για την βελτίωση της ποιότητας των συγκεκριμένων δεδομένων, μειώνοντας τις περιττές μεταδόσεις στο cloud.

(5.1 Σύνολο δεδομένων που χρησιμοποιήθηκε)

Το σύνολο δεδομένων που χρησιμοποιήθηκε είναι το Beijing Multi-Site Air-Quality Data Set, το οποίο περιλαμβάνει τα δεδομένα ανά ώρα των ατμοσφαιρικών ρύπων από 12 εθνικά ελεγχόμενους σταθμούς παρακολούθησης της ποιότητας του αέρα, από το οποίο εμείς χρησιμοποιήσαμε το σύνολο δεδομένων της πόλης Nongzhanguan. Τα δεδομένα για την ποιότητα του αέρα προέρχονται από το δημοτικό κέντρο περιβαλλοντικής παρακολούθησης του Πεκίνο. Τα μετεωρολογικά δεδομένα της τοποθεσίας ποιότητας αέρα αντιστοιχίζονται στον μετεωρολογικό σταθμό από τη μετεωρολογική υπηρεσία της Κίνας. Η χρονική περίοδος είναι από 1 Μαρτίου 2013 έως 28 Φεβρουαρίου 2017.

Οι μετρήσεις PM_{2.5} περιλαμβάνονται συχνά στις εκθέσεις για την ποιότητα του αέρα από περιβαλλοντικές αρχές και εταιρείες. Το PM_{2.5} αναφέρεται σε ατμοσφαιρικά σωματίδια (PM) που έχουν διάμετρο μικρότερη από 2,5 μικρόμετρα. Με άλλα λόγια, χρησιμοποιείται ως μέτρο ρύπανσης, όπως το CO, O₃ και το PM₁₀. Επίσης περιλαμβάνονται καιρικές παράμετροι όπως η θερμοκρασία, η πίεση, η κατεύθυνση του ανέμου, η ταχύτητα, η βροχή και η υγρασία.

Πίνακας 1–Στατιστικά Συνόλου Δεδομένων

	count	mean	std	min	25%	50%	75%	max
PM2.5	33114	84.720	85.633	2	22	59	116	844
PM10	33114	109.175	95.022	2	38	86	149	995
SO2	33114	18.560	24.109	1	3	9	23	257
NO2	33114	58.213	36.379	2	29	51	80	273
CO	33114	1327.784	1248.357	100	500	900	1600	10000
O3	33114	58.614	58.561	0.214	10	45	84	390
TEMP	33114	13.698	11.447	-16.8	3.2	14.6	23.5	41.1
PRES	33114	1012.496	10.268	987.1	1003.9	1012.1	1020.9	1042
DEWP	33114	2.521	13.789	-35.3	-8.7	3.2	15.1	28.8
RAIN	33114	0.066	0.807	0	0	0	0	46.4
WSPM	33114	1.858	1.272	0	1	1.5	2.4	10.5

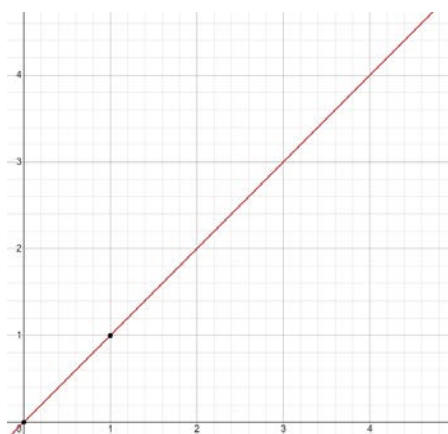
*Στρογγυλοποιημένα στα 3 δεκαδικά ψηφιά

Πίνακας 2– Χαρακτηριστικά Συνόλου δεδομένων

PM2.5	PM2.5 συγκεντρωτική τιμή σε $\mu\text{g} / \text{m}^3$
PM10	PM10 συγκεντρωτική τιμή σε $\mu\text{g} / \text{m}^3$
SO2	SO2 συγκεντρωτική τιμή σε $\mu\text{g} / \text{m}^3$
NO2	NO2 συγκεντρωτική τιμή σε $\mu\text{g} / \text{m}^3$
CO	CO συγκεντρωτική τιμή σε $\mu\text{g} / \text{m}^3$
O3	O3 συγκεντρωτική τιμή σε $\mu\text{g} / \text{m}^3$
TEMP	Θερμοκρασία σε βαθμούς Κέλσιου
PRES	Πίεση σε hPa
DEWP	Σημείο Δρόσου θερμοκρασίας σε βαθμούς Κέλσιου
RAIN	Κατακρήμνιση σε mm
WSPM	Ταχύτητα ανέμου σε m/s

(5.2 Παράμετροι πειραμάτων)

Για την διεξαγωγή πειραμάτων, αντλήσαμε δεδομένα από το παραπάνω σύνολο δεδομένων και μέσα από κάθε στήλη βρήκαμε την μέση τιμή, την τυπική απόκλιση, την μικρότερη και μεγαλύτερη τιμή. Με την βοήθεια της κανονικής κατανομής (Gaussian distribution) δημιουργήθηκαν τυχαίοι αριθμοί και τυχαία ζευγάρια αριθμών, τα οποία είναι στα όρια της μικρότερης και μεγαλύτερης τιμής. Με σκοπό την δημιουργία δυο λιστών, οι οποίες αποτελούνται από 10 κόμβους, ο κάθε κόμβος αποτελείται από 11 διαστάσεις (ο αριθμός των διαστάσεων προκύπτει από τις παραμέτρους που υπάρχουν στον Πίνακα 1) όπου κάθε περαιτέρω διάσταση περιέχει 550 στοιχεία. Στην σύγκριση μεταξύ των ζευγαριών αριθμών ο πρώτος αριθμός του ζευγαριού θα είναι μεγαλύτερος του δεύτερου ώστε το ζευγάρι να βρίσκεται στο πάνω μέρος του άξονα $y = x$.

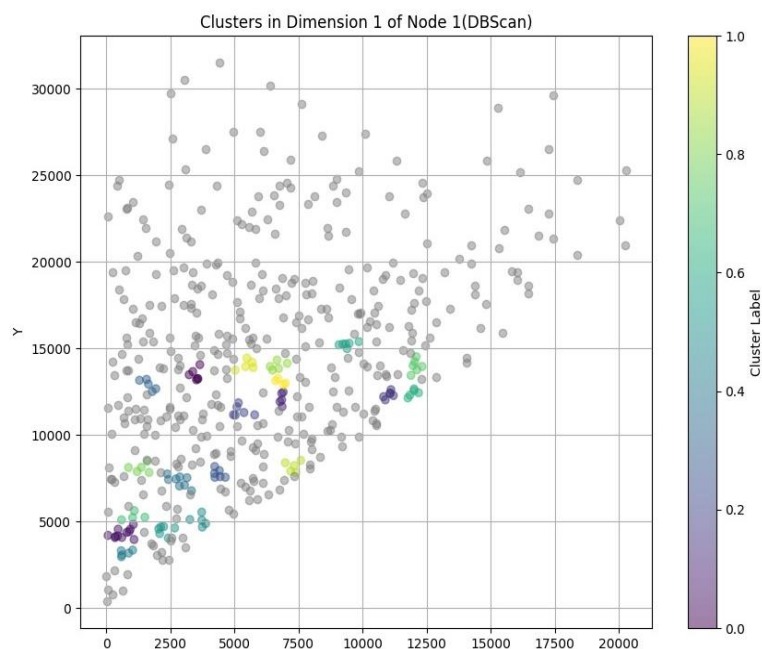


Εικόνα 16: $y=x$

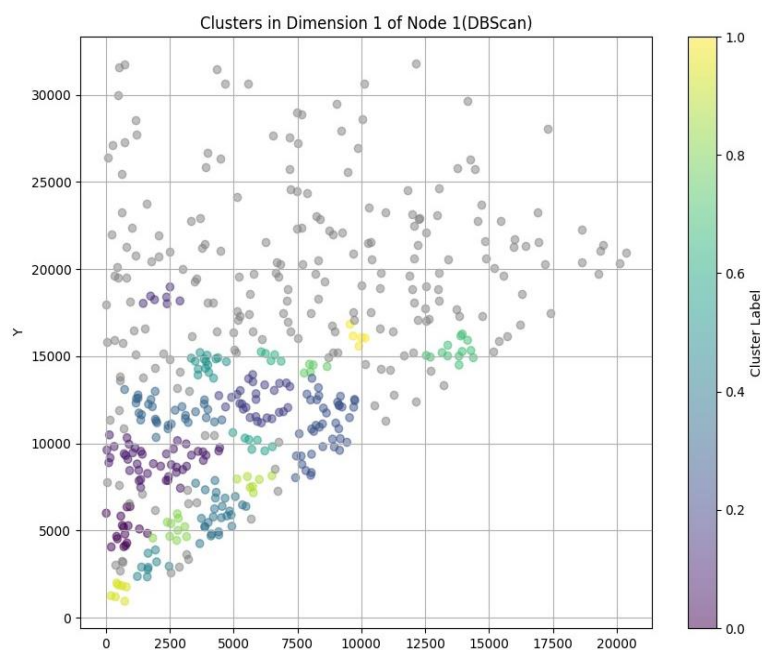
(5.3 Αποτελέσματα και αξιολόγηση πειραμάτων)

Στα παρακάτω διαγράμματα παρουσιάζονται τα τυχαία ζευγάρια αριθμών των αλγόριθμων συστάδων που χρησιμοποιήθηκαν με τις αντίστοιχες παραμέτρους τους. Τα πειράματα αυτά πραγματοποιούνται για να παρατηρήσουμε τις συστάδες από τις οποίες δημιουργούνται τα ερωτήματα καθώς και την εγγύτητα των ερωτημάτων.

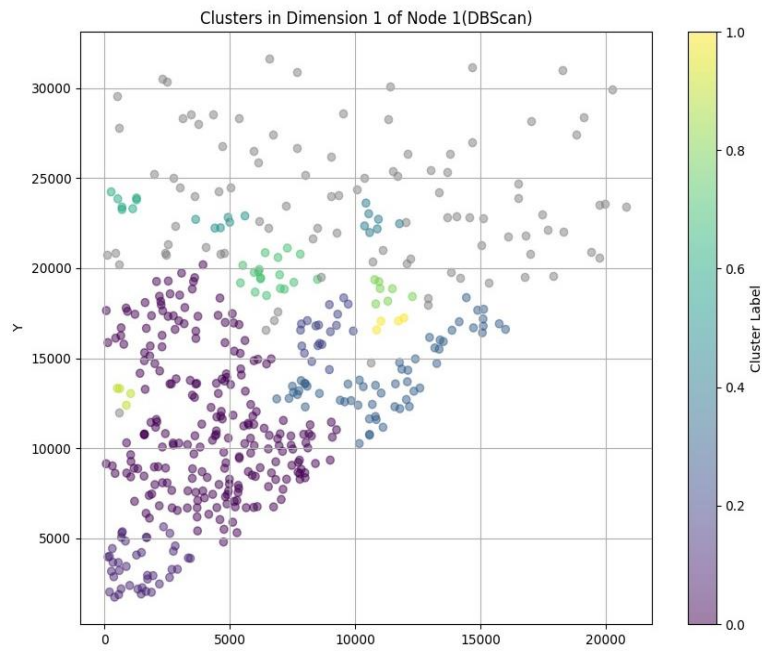
(5.3.1 Πειράματα με DBSCAN)



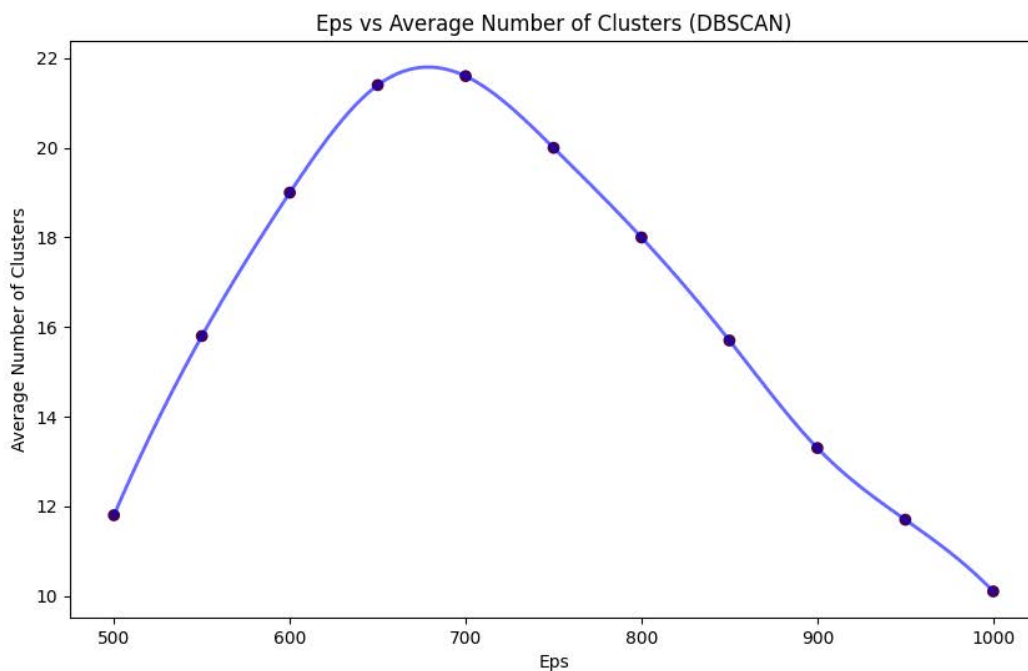
Εικόνα 17: DBSCAN(eps=500, min_samples=5)



Εικόνα 18: DBSCAN(eps=700, min_samples=5)



Εικόνα 19: DBSCAN($\epsilon=1000$, $\text{min_samples}=5$)

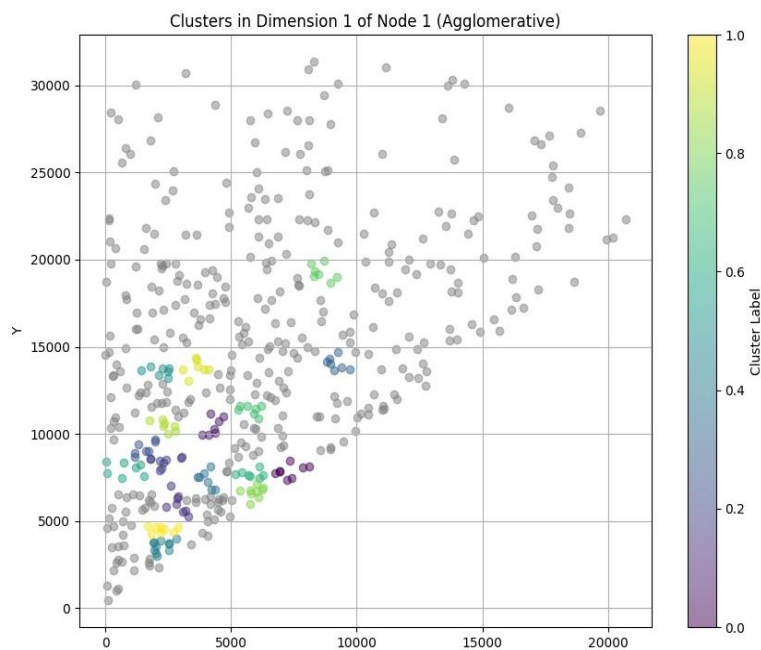


Εικόνα 20: Γραφική παράσταση DBSCAN Eps/Average number of Clusters

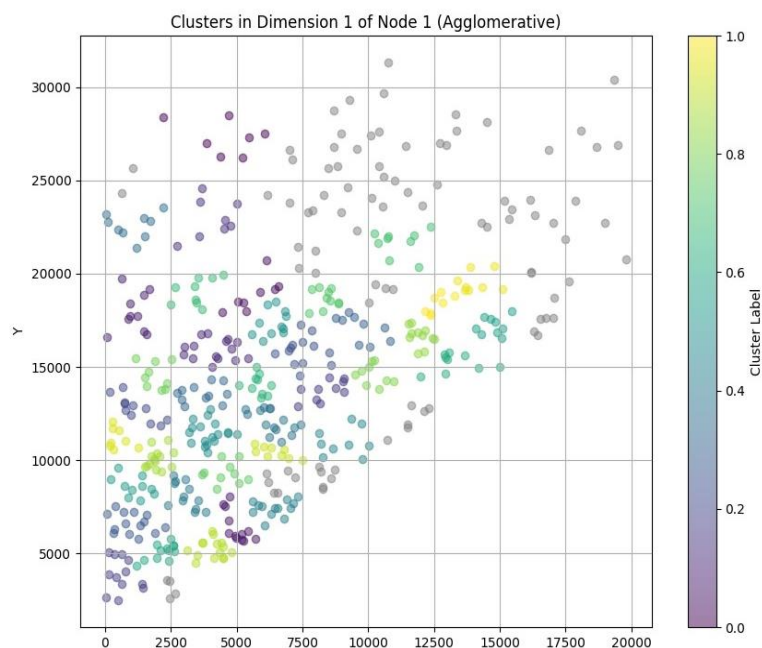
Η καμπύλη της εικόνας 20 του αλγορίθμου συστάδας DBSCAN προκύπτει από τις 11 διαφορετικές τιμές που ορίσαμε για το Eps ενώ παράλληλα αφήνουμε το min_samples σταθερό ώστε να παρατηρήσουμε αλλαγή στην μέση τιμή των συστάδων που δημιουργούνται. Παρατηρήσαμε ότι μέσω την καμπύλης αυτής ότι όσο ανεβαίνει η τιμή του Eps τόσο περισσότερες συστάδες δημιουργούνται ανά μέσο ορό μέχρι το σημείο

Eps=700 και έπειτα από το σημείο αυτό δημιουργούνται λιγότερες συστάδες οι οποίες περιέχουν περισσότερα ζευγάρια τιμών.

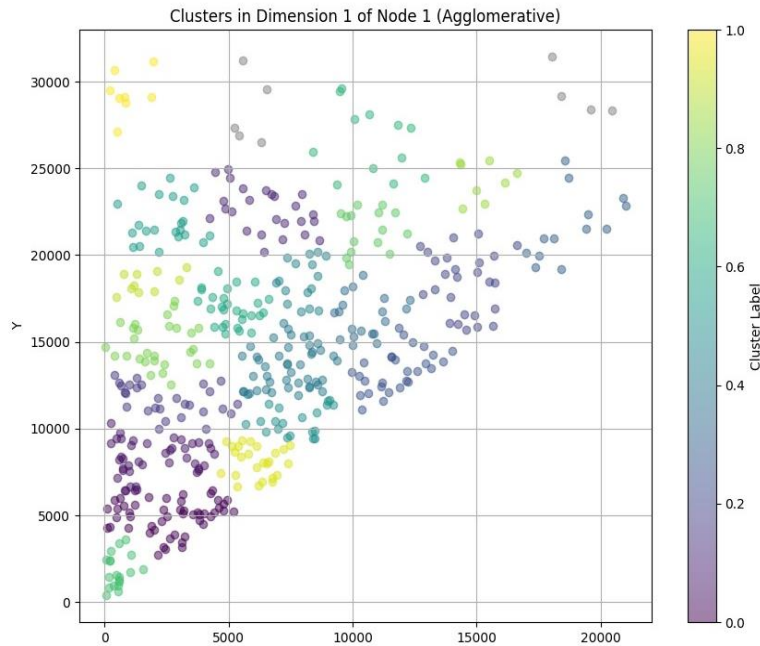
(5.3.2 Πειράματα με Agglomerative)



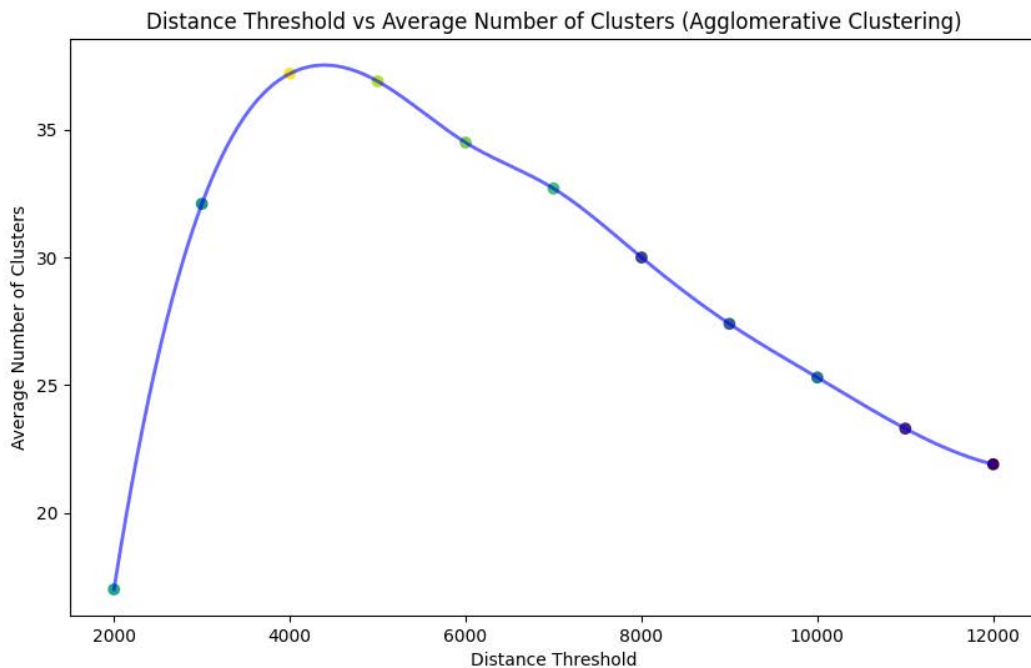
Εικόνα 21: AgglomerativeClustering(n_clusters=None, distance_threshold=2000, compute_full_tree=True)



Εικόνα 22: AgglomerativeClustering(n_clusters=None, distance_threshold=4000, compute_full_tree=True)



Εικόνα 23: AgglomerativeClustering(`n_clusters=None`, `distance_threshold=12000`, `compute_full_tree=True`)

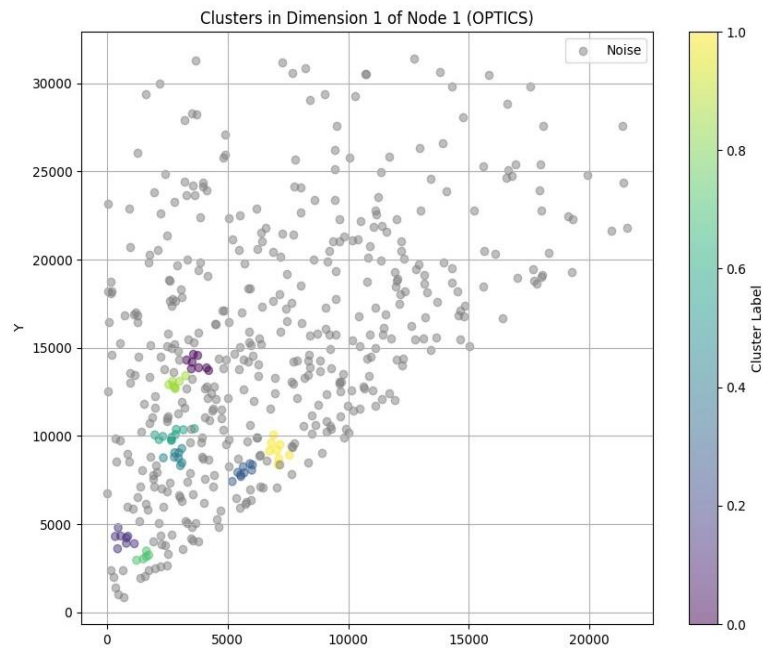


Εικόνα 24: Γραφική παράσταση Agglomerative Distance threshold/Average number of Clusters

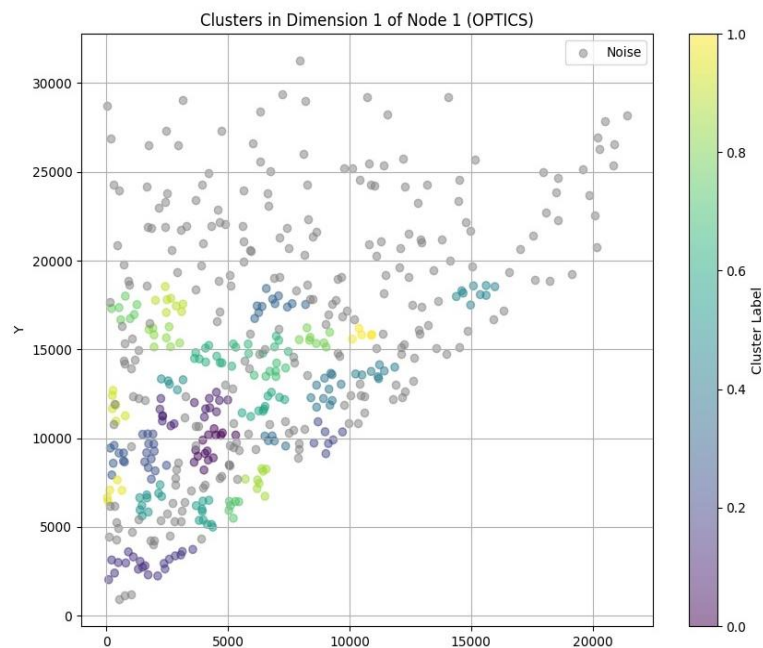
Η καμπύλη της εικόνας 24 του αλγορίθμου συστάδας Agglomerative προκύπτει από τις 11 διαφορετικές τιμές που ορίσαμε για το `distance_threshold` ενώ παράλληλα αφήνουμε το `n_clusters=None` (διότι δεν θέλουμε να έχουμε συγκεκριμένο αριθμό

συστάδων) και το `compute_full_tree=True` ώστε να παρατηρήσουμε αλλαγή στην μέση τιμή συστάδων που δημιουργούνται. Παρατηρήσαμε ότι μέσω την καμπύλης αυτής ότι από `distance_threshold=2000` έως `distance_threshold=4000` ο αριθμός των συστάδων που δημιουργούνται ανά μέσο ορό αυξάνονται, έπειτα από το σημείο αυτό δημιουργούνται λιγότερες συστάδες οι οποίες περιέχουν περισσότερα ζευγάρια τιμών.

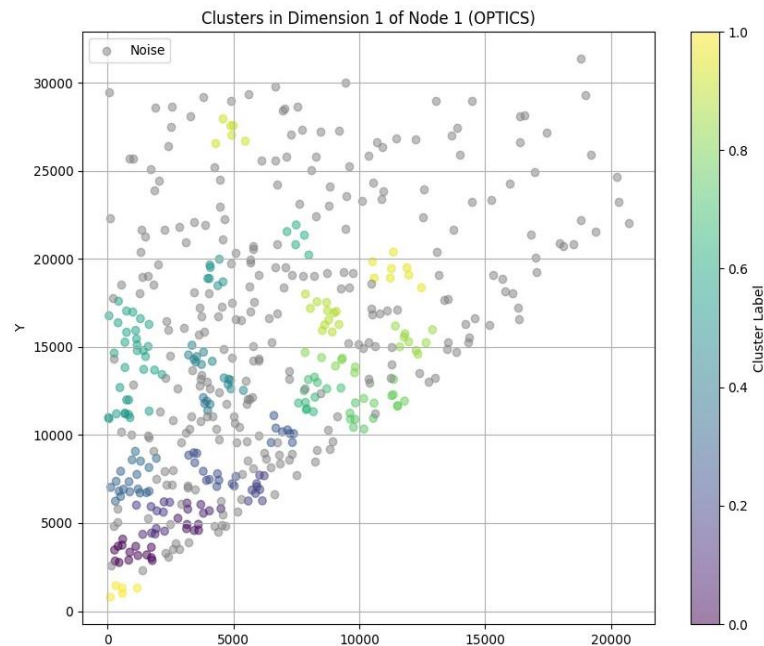
(5.3.3 Πειράματα με OPTICS)



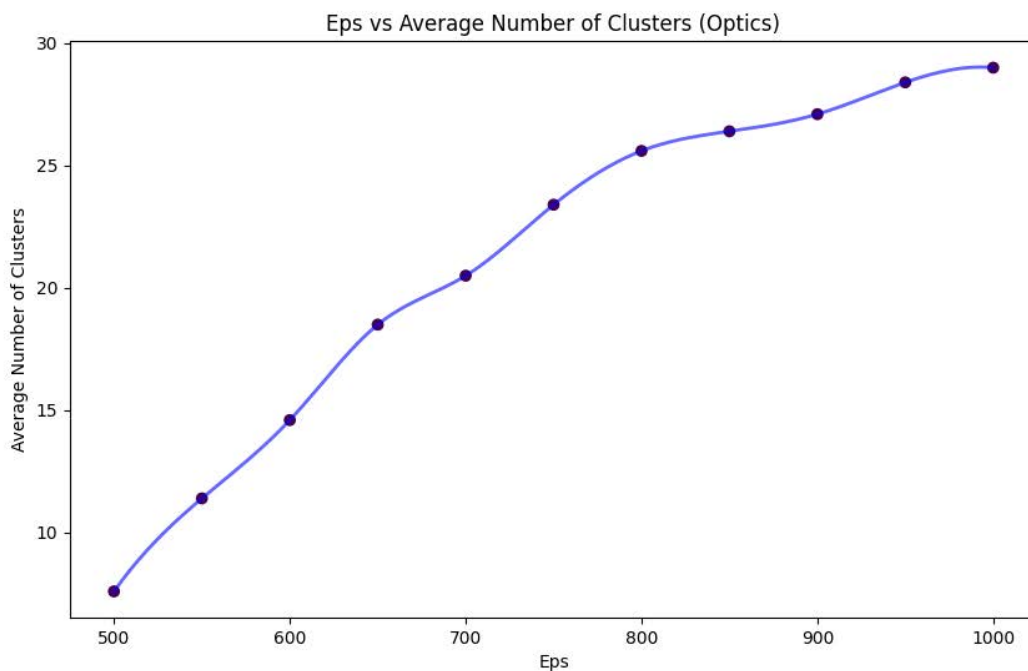
Εικόνα 25: OPTICS(min_samples=5,max_eps=500)



Εικόνα 26: OPTICS(min_samples=5,max_eps=750)



Εικόνα 27: OPTICS(min_samples=5,max_eps=1000)



Εικόνα 28: Γραφική παράσταση OPTICS Eps/Average number of Clusters

Η καμπύλη της εικόνας 28 του αλγορίθμου συστάδας OPTICS προκύπτει από τις 11 διαφορετικές τιμές που ορίσαμε για το max_eps ενώ παράλληλα αφήνουμε το min_samples σταθερό ώστε να παρατηρήσουμε αλλαγή στην μέση τιμή συστάδων που δημιουργούνται. Παρατηρήσαμε ότι μέσω την καμπύλης αυτής ότι όσο ανεβαίνει η τιμή του max_eps τόσο περισσότερες συστάδες δημιουργούνται ανά μέσο ορό μέχρι να φτάσει

σε ένα σημείο ηρεμίας και έπειτα από το σημείο αυτό δημιουργούνται παρόμοιος αριθμός συστάδων οι οποίες περιέχουν παρόμοιο αριθμό ζευγαριών τιμών. Επιπλέον, όσο μεγαλώνει ο αριθμός του max_eps τόσο μεγαλώνει ο χρόνος εκτέλεσης του προγράμματος.

(5.3.4 Υπολογισμός ποσοστού επιτυχίας)

Παράγουμε το linked_list.csv από την ήδη υπάρχουσα λίστα που δημιουργήσαμε για την ευκολότερη επεξεργασία δεδομένων, το linked_list.csv περιέχει στοιχεία όπως είναι ο κόμβος, η διάσταση και μια λίστα αριθμών στην στήλη περιεχομένων. Για κάθε εισαγωγή στο linked_list_df αναγνωρίζει τις συστάδες που ταιριάζουν συγκρίνοντας το κόμβο και την διάσταση και στα δυο πλαίσια δεδομένων. Ελέγχει πότε κάθε τιμή στην λίστα περιεχομένων πέφτει ανάμεσα στα όρια X και Y της αντίστοιχης συστάδας. Τα αποτελέσματα αυτά αποθηκεύονται σε μια λίστα η οποία έπειτα μετατρέπεται σε πλαίσιο δεδομένων Pandas. Μια επιπλέον στήλη Within_Bounds δημιουργείται από την εκτέλεση της λογικής έκφρασης “OR” ανάμεσα στις στήλες X_within_bounds και Y_within_bounds, καθορίζοντας έτσι αν μια τιμή βρίσκεται μεταξύ των ορίων X και Y. Μετράμε τον αριθμό των σειρών όπου τα δεδομένα βρίσκονται μέσα στα όρια. Ο υπολογισμός του ποσοστού επιτυχίας προκύπτει από την διαίρεση των σειρών του within_bounds με τον συνολικό αριθμό σειρών, εκφρασμένο ως ποσοστό.

Node	Dimension	Number	X_within_bounds	Y_within_bounds	Within_Bounds
1	1	20767.21561	TRUE	FALSE	TRUE
1	1	14392.53802	TRUE	FALSE	TRUE
1	1	14881.81316	TRUE	FALSE	TRUE
1	1	5070.831544	TRUE	FALSE	TRUE
1	1	9067.697258	TRUE	FALSE	TRUE
1	1	7968.114575	TRUE	FALSE	TRUE
1	1	10299.28131	TRUE	FALSE	TRUE
1	1	15381.43632	TRUE	FALSE	TRUE
1	1	7791.971517	TRUE	FALSE	TRUE
1	1	19824.25268	TRUE	FALSE	TRUE
1	1	21950.25185	TRUE	FALSE	TRUE
1	1	12535.25423	TRUE	FALSE	TRUE
1	1	5096.932149	TRUE	FALSE	TRUE
1	1	7979.457002	TRUE	FALSE	TRUE
1	1	9930.557087	TRUE	FALSE	TRUE
1	1	1317.919517	TRUE	FALSE	TRUE
1	1	21775.35174	TRUE	FALSE	TRUE
1	1	1731.243533	TRUE	FALSE	TRUE
1	1	25600.36796	TRUE	FALSE	TRUE
1	1	1976.415392	TRUE	FALSE	TRUE
1	1	2632.3603	TRUE	FALSE	TRUE

Εικόνα 29: Μέρος του checked_contents_results.csv

Τα αποτελέσματα αυτά αποθηκεύονται σε ένα νέο αρχείο CSV
checked_contents_results.csv.

(5. 3.5 Ποσοστό επιτυχίας μεταξύ κόμβων)

Cluster_No	Linked_No	Success_Rate					
1	1	80.98714	5	1	92.44637890848021	9	1
1	2	81.06988522499545	5	2	92.49430375903322	9	2
1	3	81.01460704750659	5	3	92.45707301178967	9	3
1	4	81.18548985521244	5	4	92.49164154035222	9	4
1	5	81.25848785282251	5	5	92.57711824205828	9	5
1	6	80.97488416296355	5	6	92.47547	9	6
1	7	81.06813312642079	5	7	92.42481996974355	9	7
1	8	81.14465294224813	5	8	92.46380640068014	9	8
1	9	81.19651	5	9	92.50549470609711	9	9
1	10	81.12954740650132	5	10	92.48061881338434	9	10
2	1	99.78661773079831	6	1	91.92151321018574	10	1
2	2	99.74538167243578	6	2	91.88953057630413	10	2
2	3	99.74026	6	3	91.87151000309004	10	3
2	4	99.74351999162452	6	4	91.95322182639589	10	4
2	5	99.80969483785752	6	5	91.99138	10	5
2	6	99.75507936694503	6	6	91.87152278653103	10	6
2	7	99.63279	6	7	91.81832471075776	10	7
2	8	99.70740820483944	6	8	91.90762433958827	10	8
2	9	99.77498091539712	6	9	91.90323110624315	10	9
2	10	99.72055	6	10	91.93949717617053	10	10
3	1	93.12437217138427	7	1	92.87861967088614	10	10
3	2	93.12937	7	2	92.92476771725269		
3	3	93.07596022030675	7	3	92.86817162376353		
3	4	93.20895246742725	7	4	92.95571245935952		
3	5	93.27809	7	5	93.02772697010057		
3	6	93.05505980191076	7	6	92.83503		
3	7	93.09198513858678	7	7	92.85240880934316		
3	8	93.17303091030546	7	8	92.91997327989313		
3	9	93.23060119264939	7	9	92.99696361202385		
3	10	93.11721	7	10	92.90956762008867		
4	1	85.82512903471446	8	1	92.11416547345634		
4	2	85.88445422313157	8	2	92.12750652820793		
4	3	85.86028	8	3	92.0854		
4	4	85.91310562259689	8	4	92.18376797406422		
4	5	85.96982716563211	8	5	92.19922140946142		
4	6	85.80588892651082	8	6	92.08849		
4	7	85.81352455240471	8	7	92.09429504012284		
4	8	85.88136552825932	8	8	92.11372441853405		
4	9	85.89370629370629	8	9	92.15963246927103		
4	10	85.91238140616898	8	10	92.13047914009837		

Εικόνα 30: Ποσοστά επιτυχίας όταν ο θόρυβος υπολογίζεται ως συστάδα

Cluster_N	Linked_N	Success_Rate	5	1	44.85467622966994	9	1	45.26595896866222
1	1	46.08046855569496	5	2	44.81324047873803	9	2	45.24385
1	2	46.03815905486387	5	3	45.14073730757689	9	3	45.48247668829279
1	3	46.27272	5	4	44.52811	9	4	44.98100161327967
1	4	45.77543986755642	5	5	44.37609871523001	9	5	44.79173691280145
1	5	45.56198050185776	5	6	44.936310944240844	9	6	45.371943684687004
1	6	46.18478953454338	5	7	44.48630731715261	9	7	44.93529486572103
1	7	45.658336060563244	5	8	44.90151270975622	9	8	45.25583734188345
1	8	46.016272085532606	5	9	44.153144089112466	9	9	44.522140762463344
1	9	45.32844797787764	5	10	45.055738098532615	9	10	45.42978741817451
1	10	46.21804084933472	6	1	47.8876	10	1	44.586947153059384
2	1	47.11403697812044	6	2	47.83761798475842	10	2	44.44506
2	2	46.96838468124688	6	3	48.12699439923649	10	3	44.78321055154575
2	3	47.293292243486604	6	4	47.53610364746452	10	4	44.23286175314881
2	4	46.701418145735445	6	5	47.29691395397455	10	5	44.05673555360793
2	5	46.50636437174882	6	6	47.995026114959146	10	6	44.65413243403295
2	6	47.13817539872487	6	7	47.48410923210555	10	7	44.092270422749245
2	7	46.59120913495904	6	8	47.82864652155105	10	8	44.57108956756005
2	8	47.113032958122844	6	9	47.07473948975925	10	9	43.87675225537821
2	9	46.30379658655521	6	10	48.05141864135418	10	10	44.753161638500416
2	10	47.25201906711115	7	1	44.307637928894174			
3	1	47.051535026826194	7	2	44.200886535758634			
3	2	46.91925738706328	7	3	44.49573128184079			
3	3	47.25464238268432	7	4	43.96973795223403			
3	4	46.645048760390445	7	5	43.79195			
3	5	46.45871	7	6	44.354287600259966			
3	6	47.08954418689217	7	7	43.815981598709165			
3	7	46.55087728373871	7	8	44.32902131499142			
3	8	47.03981	7	9	43.60795454545455			
3	9	46.22981818181818	7	10	44.46307511883936			
3	10	47.24510618608521	8	1	46.91699609545629			
4	1	47.25962	8	2	46.756663066637294			
4	2	47.09566875853927	8	3	47.118099209805386			
4	3	47.43174695694545	8	4	46.46896920671641			
4	4	46.80041	8	5	46.28524			
4	5	46.58931929409411	8	6	46.95237			
4	6	47.24886343353472	8	7	46.339218413380465			
4	7	46.68011053509442	8	8	46.875135058788224			
4	8	47.21838908142745	8	9	46.11530494821634			
4	9	46.41382362955397	8	10	47.027302182788205			
4	10	47.35870569876559						

Εικόνα 31: Ποσοστά επιτυχίας όταν ο δεν θόρυβος υπολογίζεται ως συστάδα

Η εικόνα 30 περιγράφει το ποσοστό επιτυχίας της σύγκρισης των παραγομένων ερωτημάτων και των κόμβων των δεδομένων όταν τα ερωτήματα έχουν παραχθεί με τον θόρυβο που παράγεται από τους αλγορίθμους συσταδοποίησης (`cluster_label= -1` παρουσιάζεται με γκρι χρώμα στα παραπάνω διαγράμματα συσταδοποίησης). Η εικόνα 31 περιγράφει το ποσοστό επιτυχίας της σύγκρισης των παραγομένων ερωτημάτων και των κόμβων των δεδομένων όταν στα ερωτήματα έχει απομακρυνθεί ο θόρυβος που παράγεται από τους αλγορίθμους συσταδοποίησης. Παρατηρούμε ότι από την μεταξύ σύγκριση των εικόνων 30 και 31, το ποσοστό επιτυχίας ανά κόμβο είναι μεγαλύτερο όταν ο θόρυβος υπολογίζεται ως μία συστάδα, διότι δημιουργούνται ερωτήματα με μεγαλύτερο εύρος, τα οποία απαντώνται από τα δεδομένα αυτά. Ενώ όταν ο θόρυβος απομακρύνεται δημιουργούνται μικρότερες ομάδες ερωτημάτων, κάνοντας πιο δύσκολη την απάντηση τους από τα δεδομένα. Με αυτόν τον τρόπο μας δίνεται η δυνατότητα να βρούμε τον κόμβο δεδομένων που απαντά πιο ικανοποιητικά στα ερωτήματα.

ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα και Μελλοντικές Προεκτάσεις

(6.1 Συμπεράσματα)

Το IoT δημιουργεί μεγάλα σύνολα δεδομένων σε καθημερινή βάση, αλλά είναι προς το παρόν θεωρούνται ως «σκοτεινά δεδομένα», δηλαδή δεδομένα που δημιουργούνται αλλά δεν αναλύονται ποτέ. Η δημιουργία μιας νέας εφαρμογής IoT πρέπει να βασίζεται στην αποτελεσματική ανάλυση του κατακλυσμού δεδομένων που δημιουργείται. Ειδικότερα, οι αλγόριθμοι συσταδοποίησης είναι υψίστης σημασίας για διεξαγωγή διερευνητικής ανάλυσης δεδομένων για τον εντοπισμό ενός συνόλου (συστάδα) παρόμοιων αντικειμένων. Οι αλγόριθμοι συσταδοποίησης είναι υπολογιστικά βαριοί σε φόρτο εργασίας και πρέπει να εκτελούνται σε συστήματα υψηλής απόδοσης υπολογιστών, ειδικά για την αντιμετώπιση μεγάλων συνόλων δεδομένων. Οι τεχνικές συσταδοποίησης είναι μέθοδοι μάθησης χωρίς επίβλεψη που περιλαμβάνουν την ομαδοποίηση σημείων δεδομένων τα οποία μπορούν να χρησιμοποιηθούν για την απόκτηση πολύτιμων γνώσεων, όπως η εξαγωγή προτύπων και ο εντοπισμός ακραίων στοιχείων μεταξύ άλλων [28].

Σε αυτήν την εργασία, εμβαθύνουμε στην ενσωμάτωση προδραστικών μοντέλων διαχείρισης δεδομένων μέσα σε περιβάλλοντα υπολογισμών ακμής στο πλαίσιο του IoT. Η έρευνά μας έχει ως στόχο την αντιμετώπιση των προκλήσεων που θέτονται από την δυναμική φύση των ροών δεδομένων αλλά και την συνεχώς αυξανόμενη ζήτηση που υπάρχει για τις δυνατότητες λήψης αποφάσεων σε πραγματικό χρόνο στους κόμβους υπολογισμών ακμής. Με την χρήση των αλγορίθμων συσταδοποίησης, παρατηρείται σημαντική βελτίωση στην αποτελεσματικότητα και την ανταπόκριση των κόμβων ακμής μέσω του προδραστικού εντοπισμού μοτίβων και ομοιοτήτων στα δεδομένα πριν την πραγματοποίηση οποιασδήποτε εκφόρτωσης εργασιών ή μετάγγιση δεδομένων. Η προδραστική αυτή προσέγγιση βελτιώνει επίσης την συνολική απόδοση και αξιοπιστία του IoT.

(6.2 Μελλοντικές Προεκτάσεις)

Έχοντας βλέψεις στις μελλοντικές προεκτάσεις, έχουμε να προτείνουμε

(6.2.1 Επεκτασιμότητα και κατανεμημένη επεξεργασία)

Επεκτασιμότητα με βάση το Cloud: καθώς οι ροές ερωτημάτων κλιμακώνονται με μέγεθος και συχνότητα, μεταβαίνοντας προς πλατφόρμες που χρησιμοποιούν υποδομές βασισμένες σε cloud όπως είναι το AWS ή το Google Cloud δίνεται η δυνατότητα βελτίωσης ανταπόκρισης του συστήματος. Με σκοπό την κατανεμημένη επεξεργασία

ερωτημάτων, διασφαλίζοντας έτσι ότι το σύστημα θα παραμείνει αποτελεσματικό ακόμη και όταν ο όγκος δεδομένων αυξάνεται.

Παραλληλισμός αναγνώρισης ερωτημάτων: εφαρμόζοντας παραλληλισμένους αλγορίθμους αναγνώρισης ερωτημάτων βελτιώνεται η ικανότητα του συστήματος να επεξεργάζεται πολλαπλές ροές ταυτόχρονα, ενώ παράλληλα μειώνεται η καθυστέρηση και βελτιώνεται η απόδοση σε περιβάλλοντα υψηλής επισκεψιμότητας.

(6.2.2 Προσαρμοζόμενα Συστήματα Μάθησης)

Αυτοβελτιωτικά Συστήματα συσταδοποίησης: Μια πολλά υποσχόμενη κατεύθυνση περιλαμβάνει την εφαρμογή μηχανισμών προσαρμοστικής μάθησης, όπου το σύστημα βελτιώνει τα μοντέλα συσταδοποίησης του καθώς επεξεργάζονται περισσότερα δεδομένα. Ενσωματώνοντας βρόχους ανάδρασης, το σύστημα θα μπορούσε να βελτιώσει την ικανότητα αναγνώρισής του με την πάροδο του χρόνου χωρίς χειροκίνητη παρέμβαση, επιτρέποντας ένα αυτόνομο σύστημα αναγνώρισης ερωτημάτων.

Διαδικτυακή μάθηση για ροές ερωτημάτων: Η ενσωμάτωση διαδικτυακών αλγορίθμων εκμάθησης επιτρέπει στο σύστημα να ενημερώνει τα μοντέλα του συνεχώς καθώς φτάνουν νέα δεδομένα, καθιστώντας το εξαιρετικά προσαρμοστικό στις αλλαγές στα μοτίβα ερωτημάτων.

Παράρτημα Κώδικες

1. Agglomerative Clustering

```
Agglomerative clustering (Exclude clusters with fewer than min_cluster_samples)
    agglomerative = AgglomerativeClustering(nclusters=None,
distance_threshold=12000, compute_full_tree=True)
    agglomerative.fit(dimension_array)
    cluster_labels_agglomerative = agglomerative.labels

    # Filter out clusters based on minimum density threshold
    unique_labels, counts = np.unique(cluster_labels_agglomerative,
return_counts=True)
    valid_clusters = unique_labels[counts >= min_cluster_samples]
```

2. DBSCAN Clustering

```
dbscan = DBSCAN(eps=700, minsamples=5)

dbscan.fit(dimension_array)
cluster_labels = dbscan.labels

#Calculate cluster bounds and save points
for cluster_label in np.unique(cluster_labels):
    cluster_points = dimension_array[cluster_labels == cluster_label]
    min_values, max_values, distance =
get_cluster_bounding_box_and_distance(cluster_points)
    node_cluster_bounds.append((dim_index, node_index, 'DBSCAN',
cluster_label, min_values, max_values, distance))
```

3. OPTICS Clustering

```
optics = OPTICS(min_samples=5,max_eps=100000)
cluster_labels_optics = optics.fit_predict(dimension_array)

# Calculate and save OPTICS cluster bounds
for cluster_label in np.unique(cluster_labels_optics):
    cluster_points = dimension_array[cluster_labels_optics == cluster_label]
    min_values, max_values, distance =
get_cluster_bounding_box_and_distance(cluster_points)
    # Write cluster points to the file
    cp_file.write(f'Node {node_index + 1}, Dimension {dim_index + 1}, Cluster
{cluster_label + 1} Points OPTICS:\n")
    points_list = get_cluster_points(cluster_points)
    for point in points_list:
```

```

cp_file.write(f"{point}\n")
cp_file.write("\n")

```

4. Combine Clusters- Συνδυασμός Συστάδων

Function to combine overlapping clusters within the same node and dimension

```

def combine_clusters(clusters):
    combined_clusters = {}

    # Iterate over each (node, dimension) key in clusters
    for key, cluster_list in clusters.items():
        node, dimension = key
        combined_clusters[(node, dimension)] = []

        # Dictionary to store merged clusters
        merged_clusters = {}

        # Iterate over each cluster in the current node-dimension combination
        for cluster in cluster_list:
            method = cluster['method']
            cluster_id = cluster['cluster_id']
            min_values = cluster['min_values']
            max_values = cluster['max_values']
            distance = cluster['distance']

            merged = False

            # Check for overlap with any previously merged clusters
            for merged_id, merged_data in merged_clusters.items():
                merged_min_values = merged_data['min_values']
                merged_max_values = merged_data['max_values']

                overlap = True
                for i in range(len(min_values)):
                    # Determine if the current cluster overlaps with the merged cluster
                    if not (min_values[i] <= merged_max_values[i] and max_values[i] >=
merged_min_values[i]):
                        overlap = False
                        break

                if overlap:
                    # Merge the current cluster with the overlapping merged cluster
                    merged_data['min_values'] = [min(merged_min_values[i], min_values[i])
for i in range(len(min_values))]

```

```

        merged_data['max_values'] = [max(merged_max_values[i],
max_values[i]) for i in range(len(max_values))]
        merged_data['methods'].add(method)
        merged = True
        break

    if not merged:
        # If no overlap, add the current cluster as a new merged cluster
        merged_clusters[cluster_id] = {
            'min_values': min_values,
            'max_values': max_values,
            'methods': {method},
            'distance': distance
        }

    # Add the merged clusters for the current node-dimension combination to the
    final result
    for merged_id, merged_data in merged_clusters.items():
        combined_clusters[(node, dimension)].append({
            'cluster_id': merged_id,
            'min_values': merged_data['min_values'],
            'max_values': merged_data['max_values'],
            'methods': list(merged_data['methods']),
            'distance': merged_data['distance']
        })

    return combined_clusters

```

5. Filtering- Φιλτράρισμα

```
import pandas as pd
import ast
import numpy as np

def main():
    # Load the CSV files
    linked_list_path = 'linked_list.csv'
    final_clusters_path = 'final_clusters.csv'

    linked_list_df = pd.read_csv(linked_list_path)
    final_clusters_df = pd.read_csv(final_clusters_path)

    # Function to parse the bounds from the string
    def parse_bounds(bounds_str):
        return ast.literal_eval(bounds_str)

    # Parse the bounds and convert them to a more usable format
    final_clusters_df['low,high(X)'] =
final_clusters_df['low,high(X)'].apply(parse_bounds)
    final_clusters_df['low,high(Y)'] =
final_clusters_df['low,high(Y)'].apply(parse_bounds)

    # Convert bounds to numpy arrays for faster computation
    final_clusters_df['low(X)'] = final_clusters_df['low,high(X)'].apply(lambda x: x[0])
    final_clusters_df['high(X)'] = final_clusters_df['low,high(X)'].apply(lambda x: x[1])
    final_clusters_df['low(Y)'] = final_clusters_df['low,high(Y)'].apply(lambda x: x[0])
    final_clusters_df['high(Y)'] = final_clusters_df['low,high(Y)'].apply(lambda x: x[1])

    # Initialize a results list
    results = []

    # Iterate over the linked list DataFrame
    for i, linked_row in linked_list_df.iterrows():
        node = linked_row['Node']
        dimension = linked_row['Dimension']
        contents = ast.literal_eval(linked_row['Contents'])

        # Filter the final_clusters DataFrame for the matching Node and Dimension
        matching_clusters = final_clusters_df[
            (final_clusters_df['Node'] == node) &
            (final_clusters_df['Dimension'] == dimension)
        ]
```

```

if matching_clusters.empty:
    continue

# Convert contents to numpy array for faster computation
contents_array = np.array(contents)

# Check each matching cluster's bounds using vectorized operations
for j, cluster_row in matching_clusters.iterrows():
    x_within_bounds = (contents_array >= cluster_row['low(X)']) &
    (contents_array <= cluster_row['high(X)'])
    y_within_bounds = (contents_array >= cluster_row['low(Y)']) &
    (contents_array <= cluster_row['high(Y)'])

    for k, number in enumerate(contents):
        results.append({
            'Node': node,
            'Dimension': dimension,
            'Number': number,
            'X_within_bounds': x_within_bounds[k],
            'Y_within_bounds': y_within_bounds[k]
        })

# Convert the results to a DataFrame
results_df = pd.DataFrame(results)

# Save the results to a new CSV file
output_path = 'checked_contents_results.csv'
results_df.to_csv(output_path, index=False)

#Do a Or operation on the X_within_bounds and Y_within_bounds columns
results_df['Within_Bounds'] = results_df['X_within_bounds'] |
results_df['Y_within_bounds']

#update the checked_contents_results.csv file
results_df.to_csv(output_path, index=False)

#count the number of rows that are within bounds
within_bounds_count = results_df['Within_Bounds'].sum()
print(f'Data within bounds: {within_bounds_count}')

#count the number of rows that are not within bounds
not_within_bounds_count = results_df.shape[0] - within_bounds_count
print(f'Data not within bounds: {not_within_bounds_count}')

```

```
# check the success rate of the filtering
total_count = results_df.shape[0]
within_bounds_count = results_df['Within_Bounds'].sum()
success_rate = within_bounds_count / total_count * 100

print(fSuccess rate: {success_rate:.2f}%)

if __name__ == "__main__":
    main()
```


BIBΛΙΟΓΡΑΦΙΑ

1. Want, R., Pering, T., & Tennenhouse, D. (2003). Comparing autonomic and proactive computing. **IBM Systems Journal**, **42**(1), 123–136.
2. Intel Research. (2002). **Proactive computing**. Intel Corporation.
3. Want, R. (2004). RFID: A key to automating everything. **Scientific American**, **290**(4), 56–65.
4. Want, R., Pering, T., & Tennenhouse, D. (2003). **Comparing autonomic and proactive computing**. Citeseerx.
<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7ffe34b14321952271fc903b3592d00a2432ae45>
5. Spiceworks. (2023). What is machine learning? Definition, types, applications, and trends. <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/>
6. Çelik, Ö., & Altunaydin, S. S. (2017). A research on machine learning methods and its applications. **Dergipark**. <https://dergipark.org.tr/en/download/article-file/532680>
7. TechTarget. (2023). **Top 12 machine learning use cases and business applications**. <https://www.techtarget.com/searchenterpriseai/feature/Top-12-machine-learning-use-cases-and-business-applications>
8. Muneer, A., Kumar, A., Khurana, A., & Kumawat, S. (2018). A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification. **IEEE Xplore**.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8663155>
9. Van Otten, N. (2023, May 2). **Regression vs classification: Understand how to choose and switch between them**. Spot Intelligence.
<https://spotintelligence.com/2023/05/02/regression-vs-classification/>
10. Serra, A., & Tagliaferri, R. (2017). Unsupervised learning: Clustering. **ResearchGate**. https://www.researchgate.net/profile/Angela-Serra-3/publication/322543367_Unsupervised_Learning_Clustering/links/5a6f7680458515015e616155/Unsupervised-Learning-Clustering.pdf
11. Yadav, A., & Mittal, M. (2016). A survey on unsupervised clustering algorithm based on K-means clustering. **ResearchGate**.
https://www.researchgate.net/profile/Ashish-Yadav-36/publication/311843914_A_Survey_on_Unsupervised_Clustering_Algorithm_base_d_on_K-Means_Clustering/links/585ccfcc08ae8fce48fbad75/A-Survey-on-Unsupervised-Clustering-Algorithm-based-on-K-Means-Clustering.pdf

12. Plaza, A., Martínez, P., Plaza, J., & Pérez, R. (2005). Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. **IEEE Transactions on Geoscience and Remote Sensing**, **43**(3), 433–441.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1396320>
13. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. **Journal of Artificial Intelligence Research**, **4**, 237–285.
<https://www.jair.org/index.php/jair/article/view/10166/24110>
14. Williams, B. (2019). **Machine learning: Evaluating classification models**. Medium. <https://medium.com/@brandon93.w/machine-learning-evaluating-classification-models-18713af3d764>
15. Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A survey on the edge computing for the internet of things. **IEEE Xplore**.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8123913>
16. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. **IEEE Internet of Things Journal**, **3**(5), 637–646.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7488250>
17. Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. **Future Generation Computer Systems**, **97**, 219–235.
<https://www.sciencedirect.com/science/article/pii/S0167739X18319903>
18. Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. **IEEE Access**, **8**, 85714–85735.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9083958>
19. Heck, M., Edinger, J., Schafer, D., & Becker, C. (2018). IoT applications in fog and edge computing: Where are we and where are we going? **IEEE Xplore**.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8487455>
20. Puschmann, D., Barnaghi, P., & Tafazolli, R. (2016). Adaptive clustering for dynamic IoT data streams. **IEEE Xplore**.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7593263>
21. Hassan, M. M., Eesa, A. S., Mohammed, A. J., & Arabo, W. K. (2019). Oversampling method based on Gaussian distribution and K-means clustering. **CMC-Computers, Materials & Continua**, **69**(1), 29–46.
https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC_69-1/TSP_CMC_18280/TSP_CMC_18280.pdf
22. Ehsan Jolous Jamshidi, Yusri Yusup, John Stephen Kayode, Mohamad Anuar Kamaruddin, Detecting outliers in a univariate time series dataset using

unsupervised combined statistical methods: A case study on surface water temperature, *Ecological Informatics*, Volume 69,2022,101672,ISSN 1574-9541,<https://www.sciencedirect.com/science/article/pii/S1574954122001224>

23. Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. L., ElKhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. **IEEE Xplore**.
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8713992>

24. Kolomvatsos, K., & Anagnostopoulos, C. (2023). Autonomous proactive data management in support of pervasive edge applications. **Future Generation Computer Systems**, **144**, 155–166.
<https://www.sciencedirect.com/science/article/pii/S0167739X24000487#sec3>

25. Bouguettaya, A., Yu, Q., Liu, X., Zhou, X., & Song, A. (2014). Efficient agglomerative hierarchical clustering. **Expert Systems with Applications**, **41*(11)*, 5849–5860.
<https://www.sciencedirect.com/science/article/pii/S0957417414006150>

26. Ram, A., Jalal, S., Jalal, A. S., & Kumar, M. (2015). A density-based algorithm for discovering density varied clusters in large spatial databases. **ResearchGate**.
https://www.researchgate.net/publication/282334683_A_Density_based_Algorithm_for_Discovering_Density_Varied_Clusters_in_Large_Spatial_Databases

27. Schubert, E., & Gertz, M. (2018). Improving the cluster structure extracted from OPTICS plots. **RWTH Aachen University**. <http://star.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-2191/paper37.pdf>

28. Cecilia, J. M., Cano, J. C., Morales-García, J., Llanes, A., & Imbernón, B. (2020). Evaluation of clustering algorithms on GPU-based edge computing platforms. **Sensors**, **20*(20)*, 6335.

MathWorks. (n.d.). Reinforcement learning. MathWorks.

<https://www.mathworks.com/discovery/reinforcement-learning.html>

Simplilearn. (2023). Regression vs classification in machine learning. Simplilearn.

<https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article>

Primo AI. (n.d.). Unsupervised learning. Primo AI.

<https://primo.ai/index.php?title=Unsupervised>

Spiceworks. (2023). What is reinforcement learning? Spiceworks.

<https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-reinforcement-learning/>

Tridens Technology. (2023). All about the Internet of Things (IoT). Tridens

Technology. <https://tridenttechnology.com/all-about-the-internet-of-things-iot/>

Spiceworks. (2023). What is cloud computing? Spiceworks.

<https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/>

Wipro. (2023). Edge computing: Understanding the user experience. Wipro.

<https://www.wipro.com/infrastructure/edge-computing-understanding-the-user-experience/>

Innominds. (2023). Edge computing-driven transformation of data management

applications. Innominds. <https://www.innominds.com/blog/edge-computing-driven-transformation-of-data-management-applications>

Towards Data Science. (2020). Scaling agglomerative clustering for big data: An

introduction to RAC. Towards Data Science. <https://towardsdatascience.com/scaling-agglomerative-clustering-for-big-data-an-introduction-to-rac-fb26a6b326ad>

AIMind. (2023). DBSCAN algorithm explained easily. AIMind.

<https://pub.aimind.so/dbscan-algorithm-explained-easily-170d57ca88da>

Scribbr. (2023). Normal distribution. Scribbr.

<https://www.scribbr.com/statistics/normal-distribution/>