

Ιόνιο Πανεπιστήμιο - Τμήμα Πληροφορικής



Αξιοποίηση γεωαναφορών ιχνηλάτησης σκαφών για την αυτοματοποιημένη αναγνώριση και οριοθέτηση λιμένων και ελεύθερων αγκυροβολίων με εφαρμογή τεχνικών μηχανικής μάθησης

Exploiting vessels automatic identification system geolocated tracks for smart identification and demarcation of ports and anchorages based on machine learning techniques

**Τζιάνας Νικόλαος
Π2014107**

Επιβλέπων Καθηγητής: Δημήτριος Ρίγγας

Σεπτέμβριος 2021

Πτυχιακή εργασία

Αξιοποίηση γεωαναφορών ιχνηλάτησης σκαφών για την αυτοματοποιημένη αναγνώριση και οριοθέτηση λιμένων και ελεύθερων αγκυροβολίων με εφαρμογή τεχνικών μηχανικής μάθησης

Τζιάνας Νικόλαος
Π2014107

Τριμελής εξεταστική επιτροπής

Ρίγγας Δημήτριος

Κερμανίδου Κάτια-Λήδα

Στεφανιδάκης Μιχαήλ

ΕΔιΠ

Αναπληρώτρια Καθηγήτρια

Αναπληρωτής Καθηγητής

Ημερομηνία παρουσίασης: 24/9/2021

(κενή σελίδα)

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κ. Δημήτριο Ρίγγα για την καθοδήγηση του καθ' όλη τη διάρκεια της πτυχιακής και για την δουλειά του ως καθηγητής στο Ιόνιο Πανεπιστήμιο καθώς ήταν η πρώτη μου σοβαρότερη επαφή με τον προγραμματισμό. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου που στάθηκε στο πλευρό μου από την αρχή των σπουδών μου μέχρι σήμερα. Τέλος, θα ήθελα να ευχαριστήσω την VesselBot που παρείχε τα δεδομένα που χρειάστηκαν για την εκπόνηση της εργασίας και που χωρίς την συνεισφορά τους αυτή η πτυχιακή δεν θα ήταν δυνατή.

Περίληψη

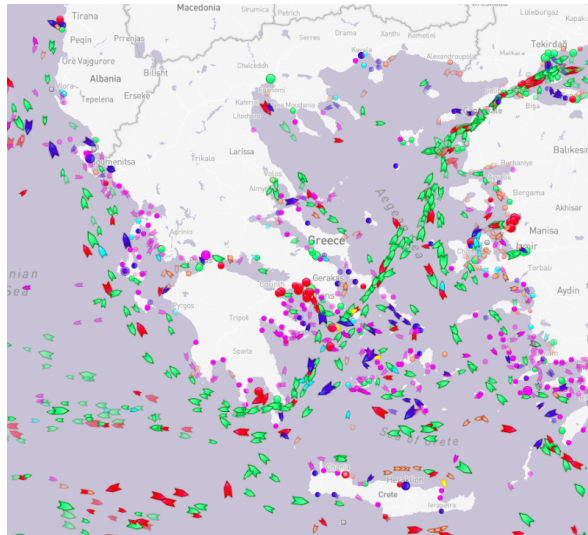
Την τελευταία δεκαετία, το αυτόματο σύστημα αναγνώρισεως έχει γίνει σημαντική πηγή πληροφορίας για την δημιουργία στατιστικών δεδομένων και την ανάπτυξη έξυπνων συστημάτων στον τομέα της ναυτιλίας. Ένα από τα προβλήματα του τομέα σήμερα, είναι ο ακριβής ορισμός των συνεχώς μεταβαλλόμενων ορίων των λιμανιών για την παρακολούθηση των πλοίων και των κινήσεων τους. Στην συγκεκριμένη εργασία, συγκρίνουμε διαφορετικούς αλγόριθμους ανάλυσης συστάδων και τα αποτελέσματα που έχουν για τον βέλτιστο εντοπισμό λιμένων και την οριοθέτηση τους με την χρήση των δεδομένων του αυτόματου συστήματος αναγνώρισεως. Ο αλγόριθμος μας εντοπίζει και οριοθετεί με επιτυχία όλα τα μεγαλύτερα λιμάνια της Ελλάδας καθώς και περισσότερα από τα μισά μικρότερα λιμάνια.

1. Εισαγωγή

1.1 Το Αυτόματο Σύστημα Αναγνώρισεως

Το Αυτόματο Σύστημα Αναγνώρισεως, ευρύτερα γνωστό ως AIS(Automatic Identification System) , είναι ένα αυτόνομο σύστημα παρακολούθησης πλοίων που αναπτύχθηκε με κύριο στόχο την αποφυγή συγκρούσεων. Το AIS χρησιμοποιεί πομπό και δέκτη για να μεταδίδει αυτόματα μηνύματα που περιέχουν πληροφορίες για το πλοίο όπως την θέση του, τον προορισμό του, την ταχύτητα του κ.α. ανά μερικά δευτερόλεπτα. Βάσει του διεθνούς οργανισμού ναυσιπλοΐας (International Maritime Organization - IMO) όλα τα πλοία με ολική χωρητικότητα μεγαλύτερη των 299 τόνων καθώς και κάθε επιβατηγό καράβι ανεξαρτήτως μεγέθους υποχρεούνται να διαθέτουν πομπό AIS.

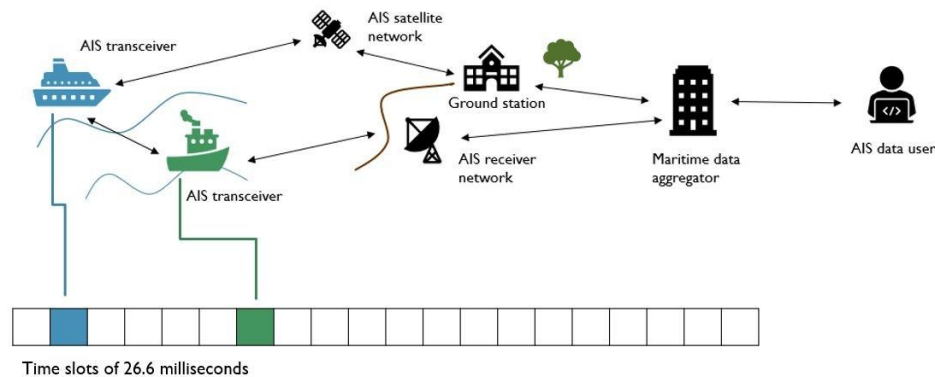
Κίνηση των πλοίων στην Ελλάδα βάσει του AIS



(<https://www.marinetraffic.com>)

Αν και όπως αναφέρθηκε η αρχική χρήση του AIS ήταν η αποφυγή συγκρούσεων , ο μεγάλος όγκος των δεδομένων που παράγει είναι δυνητικά σημαντική πηγή πληροφορίας στην εποχή των Big Data. Ένα από τα μεγαλύτερα εμπόδια για την αξιοποίηση τους ξεπεράστηκε το 2009, όταν διάφορες εταιρείες όπως η exactEarth, Spire, Spacequest και κάποια κυβερνητικά προγράμματα πειραματίστηκαν με δέκτες AIS σε δορυφόρους και κατάφεραν για πρώτη φορά να συγκεντρώσουν real-time δεδομένα για καράβια από όλον τον κόσμο.

Η διαδικασία συλλογής δεδομένων AIS



Η διαδικασία συλλογής δεδομένων AIS: Τα δεδομένα συλλέγονται από δέκτες AIS σε δορυφόρους και λιμάνια και επεξεργάζονται σε ένα κέντρο συλλογής θαλάσσιων δεδομένων πριν φτάσουν στον αναλυτή. Η μετάδοση των δεδομένων AIS γίνεται με time slots των 26.6ms για την αποφυγή συγκρούσεων κατά την αποστολή τους.

(<https://unstats.un.org/wiki/display/AIS/Introduction>)

Τα συγκεντρωτικά δεδομένα του συστήματος μπορούν να χρησιμοποιηθούν για την ταυτόχρονη παρακολούθηση της κίνησης των καραβιών ανά τον κόσμο σε πραγματικό χρόνο. Ο όγκος και η πληρότητα αυτών των δεδομένων επιτρέπει την ανάπτυξη πολύπλοκων πληροφοριακών συστημάτων για την υποστήριξη του τομέα της ναυτιλίας, αλλά οι εφαρμογές του AIS δεν περιορίζονται εκεί. Στην δημοσίευση των Yang et al. (2019, pp. 7-12) γίνεται μία αναθεώρηση των διαφορετικών δημοσιεύσεων που αφορούν το αυτόματο σύστημα αναγνώρισης και γίνεται η ταξινόμηση τους σε τρεις διαφορετικές κατηγορίες ανάλογα με τους στόχους τους:

- **Βασικές εφαρμογές:** data mining, ασφάλεια πλοήγησης
- **Εκτεταμένες εφαρμογές:** Συμπεριφορά πλοίων (ανάλυση αλιείας, ανάλυση δρομολογίων κ.α) , περιβαλλοντικές εφαρμογές (εκτίμηση εκπομπών CO₂, ανάλυση ρίσκου πετρελαιοκηλίδων κ.α)
- **Προηγμένες εφαρμογές:** Ανάλυση εμπορίου, αποδοτικότητα πλοίων και λιμένων, αρκτική ναυτιλία

Βασικές εφαρμογές όπως το data mining στα ακατέργαστα δεδομένα AIS είναι απαραίτητες καθώς επιτρέπουν την εξαγωγή πληροφορίας που μπορεί να χρησιμοποιηθεί ως βάση σε περαιτέρω αναλύσεις με εκτεταμένες ή προηγμένες εφαρμογές.

1.2 Στόχος της εργασίας

Στόχος της εργασίας είναι να προσθέσει στο κομμάτι της επεξεργασίας ακατέργαστων δεδομένων AIS, αξιολογώντας τα αποτελέσματα διάφορων αλγορίθμων μηχανικής μάθησης πάνω στο πρόβλημα του εντοπισμού και της οριοθέτησης θαλασσιών γεωγραφικών περιοχών. Η παρακολούθηση των πλοίων μέσω των δεδομένων AIS έχει ως απαραίτητη προϋπόθεση την πληροφορία της τοποθεσίας των λιμανιών και των ορίων τους. Στην ανάλυση δεδομένων, η ακρίβεια είναι ένα από τα σημαντικότερα πράγματα. Καλά ορισμένα γεωγραφικά όρια στα λιμάνια μπορούν να επιτρέψουν σε ένα πληροφοριακό σύστημα να δώσει έγκυρα πραγματικού χρόνου δεδομένα για τα port calls ανά τον κόσμο - την στιγμή άφιξης και αναχώρησης ενός καραβιού σε ένα λιμάνι. Για την παρουσίαση του πως είναι δυνατό να αναπτυχθεί ένα τέτοιο πληροφοριακό σύστημα γίνεται η υλοποίηση ενός προγράμματος παρακολούθησης πλοίων με βάση τα αποτελέσματα των αλγορίθμων μας.

1.3 Τα δεδομένα του συστήματος

Ένα AIS μεταδίδει πληροφορίες μέσω ραδιοσήματος με την μορφή κωδικοποιημένων μηνυμάτων. Τα μηνύματα χωρίζονται σε 27 διαφορετικούς τύπους ανάλογα με τις πληροφορίες (πεδία) που περιέχουν και την συχνότητα μετάδοσης τους. Κάποια πεδία του μηνύματος υπολογίζονται αυτόματα από το σύστημα (π.χ τοποθεσία, ταχύτητα) ενώ άλλα πρέπει να συμπληρωθούν χειροκίνητα από το πλήρωμα (π.χ προορισμός). Αυτός ο διαχωρισμός είναι ιδιαίτερα σημαντικός καθώς τα δεδομένα που συμπληρώνονται χειροκίνητα θεωρούνται λιγότερο αξιόπιστα λόγω του ανθρώπινου παράγοντα που τα κάνει πιο επιρρεπή σε σφάλματα. Ο Harati-Mokhtari (2006), αν και δεν κάνει άμεση σύγκριση με τα πεδία που δημιουργούνται αυτόματα από το σύστημα, αναφέρεται στα διάφορα προβλήματα που υπάρχουν με την αξιοπιστία των AIS δεδομένων που συμπληρώνονται χειροκίνητα. Ο παρακάτω πίνακας περιγράφει κάποια από τα σημαντικότερα πεδία για την ανάλυση δεδομένων, τους τύπους μηνυμάτων που τα περιέχουν και τον τρόπο με τον οποίο συμπληρώνονται:

Πεδίο	Περιγραφή	Τύποι Μηνύματος	Συμπληρώνεται
msg_type	Ο τύπος του μηνύματος	Όλα	Αυτόματα
timestamp	ISO 8601 timestamp την στιγμή της αναμετάδοσης	Όλα	Αυτόματα
mmsi	Μοναδικό αναγνωριστικό πλοίου	Όλα	Χειροκίνητα
latitude	Γεωγραφικό πλάτος σε μοίρες	1,2,3,4,18,19,27	Αυτόματα
longitude	Γεωγραφικό μήκος σε μοίρες	1,2,3,4,18,19,27	Αυτόματα
destination	Προορισμός του πλοίου	5	Χειροκίνητα
eta	Προβλεπόμενη ώρα άφιξης σύμφωνα με τον καπετάνιο, ISO 8601	5	Χειροκίνητα
draught	Το βύθισμα του πλοίου σε δέκατα	5	Χειροκίνητα
speed	Η ταχύτητα του πλοίου σε κόμβους	1,2,3,18,19,27	Αυτόματα
status	Η κατάσταση πλοήγησης του πλοίου π.χ Moored, Anchored.	1, 2, 3, 18, 19	Χειροκίνητα
ship_and_cargo_type	Ο τύπος του πλοίου	5,19,24	Χειροκίνητα

Φυσικά αυτό δεν σημαίνει πως τα δεδομένα που παράγει αυτόματα το σύστημα είναι πλήρως αξιόπιστα. Ο βασικός στόχος του συστήματος ακόμα και σήμερα είναι να προειδοποιεί άλλα πλοία που βρίσκονται στην εμβέλεια - όχι να παράγει δεδομένα για την δημιουργία στατιστικών. Αυτό σημαίνει πως περιστασιακά σφάλματα θεωρούνται αποδεκτά για τους βασικούς στόχους του συστήματος. Για παράδειγμα, είναι γνωστό πως καράβια που μεταφέρουν εμπορεύματα συνηθίζουν να κλείνουν τον AIS πομπό τους όταν διασχίζουν περιοχές με έντονη πειρατική ενέργεια για να αποκρύψουν την τοποθεσία τους. Άλλες φορές, μπορεί να υπάρξει σφάλμα κατά την μετάδοση του ραδιοσήματος λόγω κακοκαιρίας. Το σημαντικό είναι πως η εγκυρότητα και η πληρότητα των δεδομένων δεν είναι εγγυημένη και η αξιολόγηση της ποιότητας τους είναι ένα σημαντικό βήμα για την χρήση τους για οποιαδήποτε ανάλυση.

1.4 Δομή της υπόλοιπης εργασίας

Στο κεφάλαιο 2 γίνεται μία αναφορά στην βιβλιογραφία του εντοπισμού και της οριοθέτησης λιμένων και αναλύεται η μεθοδολογία της εργασίας. Επίσης γίνεται μια βασική σύγκριση των αλγορίθμων ανάλυσης συστάδων k-means, DBSCAN και OPTICS και των χαρακτηριστικών τους. Το κεφάλαιο 3 περιλαμβάνει τον σχεδιασμό των προγραμμάτων που υλοποιήθηκαν και παρουσιάζει τις επιλογές που έγιναν για τις τεχνολογίες που χρησιμοποιήθηκαν και τον λόγο της επιλογής τους. Το κεφάλαιο 4 περιγράφει την διαδικασία υλοποίησης της ανάλυσης της εργασίας και τις επιλογές που έγιναν κατά την εκτέλεση των αλγορίθμων. Το κεφάλαιο 5 αφορά την υλοποίηση του προγράμματος παρακολούθησης πλοίων με την χρήση των πολυγώνων που προέκυψαν από το τέταρτο κεφάλαιο. Το κεφάλαιο 6 αξιολογεί τα αποτελέσματα της εργασίας. Τέλος, το κεφάλαιο 7 παρουσιάζει τα γενικότερα συμπεράσματα από την διαδικασία εκπόνησης της εργασίας και πιθανές μελλοντικές επεκτάσεις στην εργασία που μπορούν να βελτιώσουν τα αποτελέσματα της.

Στο appendix βρίσκονται τα σημαντικότερα προγράμματα που υλοποιήθηκαν για την ανάλυση δεδομένων της εργασίας.

2. Σχετικές Εργασίες & Μεθοδολογία

2.1 Μηχανική Μάθηση

Η ανάλυση δεδομένων της εργασίας γίνεται με την χρήση αλγορίθμων μηχανικής μάθησης. Σύμφωνα με την IBM (2020), "Η μηχανική μάθηση είναι ένας κλάδος της τεχνητής νοημοσύνης και της πληροφορικής που ασχολείται με την χρήση δεδομένων και αλγορίθμων που μιμούνται τον τρόπο με τον οποίο μαθαίνουν οι άνθρωποι, βελτιώνοντας την ακρίβεια τους"

Η ανάγκη για την χρήση της μηχανικής μάθησης προέρχεται από την συνεχώς αυξανόμενη πολυπλοκότητα των προβλημάτων της πληροφορικής. Ένας κλασικός αλγόριθμος λύνει προβλήματα βάσει μιας σειράς απλών βημάτων που ορίζονται από κάποιον προγραμματιστή. Για κάποια προβλήματα όμως, τα βήματα που απαιτούνται για την λύση του είναι υπερβολικά πολύπλοκα για να κατανοηθούν και να οριστούν από έναν άνθρωπο. Η μηχανική μάθηση επιτρέπει την ανάπτυξη αλγορίθμων που χρησιμοποιούν δεδομένα για να βελτιώσουν τον εαυτό τους και μπορούν να ακολουθούν βήματα για τα οποία δεν έχουν ρητά προγραμματιστεί.

Συνήθως ένας αλγόριθμος μηχανικής μάθησης μπορεί να ταξινομηθεί σε μία από δύο ευρύς κατηγορίες ανάλογα με τα δεδομένα που δέχεται:

- **Επιβλεπόμενη Μάθηση:** Ο αλγόριθμος δέχεται τα δεδομένα εισόδου και τα αποτελέσματα τους και προσπαθεί να δημιουργήσει ένα μοντέλο πρόβλεψης της εξόδου, βάσει των δεδομένων εισόδου. π.χ Δημιουργία μοντέλου ταξινόμησης φωτογραφιών φρούτων. Ο αλγόριθμος εκπαιδεύεται με την χρήση ενός συνόλου φωτογραφιών για κάθε φρούτο. Μετά από αρκετά δεδομένα μπορεί να εντοπίσει και να αναγνωρίσει τα χαρακτηριστικά που ξεχωρίζουν το ένα φρούτο από το άλλο σε μια φωτογραφία.
- **Μη Επιβλεπόμενη Μάθηση:** Ο αλγόριθμος δεν δέχεται τα επιθυμητά δεδομένα εξόδου. Στόχος του είναι η αναγνώριση κρυμμένων προτύπων στα δεδομένα χωρίς την ανθρώπινη παρέμβαση. π.χ Εντοπισμός κατηγοριών πελατών από τα δεδομένα πωλήσεων ενός καταστήματος. Ο αναλυτής δεν γνωρίζει πόσες και ποιες κατηγορίες θα δημιουργηθούν πριν την εκτέλεση του αλγορίθμου.

2.2 Ανάλυση Συστάδων

Όλες οι τεχνικές μηχανικής μάθησης που θα χρησιμοποιηθούν στην εργασία εντάσσονται στην ανάλυση συστάδων. Η ανάλυση συστάδων είναι μία από τις συνηθέστερες εργασίες μη επιβλεπόμενης μάθησης και εκτελείται όταν θέλουμε να χωρίσουμε ένα σύνολο αντικειμένων ή παρατηρήσεων σε διαφορετικές ομάδες βάσει της ομοιότητας τους. Για την ανάλυση συστάδων δύο αντικείμενα θεωρούνται παρόμοια όταν οι τιμές των χαρακτηριστικών τους είναι κοντά μεταξύ τους. Αυτό μπορεί να οπτικοποιηθεί με την παρακάτω εικόνα, όπου ένα σύνολο πελατών με χαρακτηριστικά το εισόδημα και τα έξοδα τους κατηγοριοποιούνται σε διάφορες κατηγορίες πελατών:

Ανάλυση συστάδων στα δεδομένα πελατών



(<https://www.kdnuggets.com/2019/09/hierarchical-clustering.html>)

Ο αλγόριθμος δεν γνωρίζει τι αντιπροσωπεύει η κάθε κατηγορία. Η επισήμανση των συστάδων γίνεται από τον αναλυτή και βασίζεται στην γνώσεις του για τον τομέα. Επειδή η ανάλυση συστάδων μπορεί να χρησιμοποιηθεί για την εξαγωγή προτύπων σε οποιαδήποτε δεδομένα, οι πιθανές εφαρμογές της είναι πολλές.

Ένα παράδειγμα χρήσης της ανάλυσης συστάδων φαίνεται από τους Erman, Arlitt & Mahanti(2006), όπου χρησιμοποίησαν αλγορίθμους ανάλυσης συστάδων και δεδομένα από το επίπεδο μεταφοράς για την ταξινόμηση της κίνησης ενός δικτύου με αρκετά καλά αποτελέσματα. Στόχος της ταξινόμησης της κίνησης του δικτύου είναι η ομαδοποίηση και αναγνώριση των διαφορετικών κατηγοριών κίνησης που δέχεται μία ιστοσελίδα με σκοπό την καλύτερη διαχείρισή τους.

2.3 Σχετικές Πηγές

Ο εντοπισμός και η οριοθέτηση λιμένων θα επιτευχθεί με την χρήση αλγορίθμων μηχανικής μάθησης και πιο συγκεκριμένα ανάλυσης συστάδων. Παρόμοιες δημοσιεύσεις μπορούν να χρησιμοποιηθούν ως σημείο αναφοράς και πιθανώς να βελτιώσουν την μεθοδολογία της εργασίας.

Δυστυχώς, λόγω της εξειδίκευσης του θέματος της εργασίας, οι δημοσιεύσεις με τον ίδιο στόχο είναι περιορισμένες. Υπάρχουν εταιρείες που έχουν καταφέρει να δημιουργήσουν πολύγωνα αντιπροσωπευτικά των λιμανιών (Oceanbolt, Marinetraffic) , αλλά οι μεθοδολογίες τους δεν είναι δημόσια διαθέσιμες. Αναφορά θα γίνει σε δύο πηγές με παρόμοια μεθοδολογία μεταξύ τους που μπορούν να χρησιμοποιηθούν ως βάση για την υλοποίηση της εργασίας.

Η πρώτη πηγή είναι ένα άρθρο της GA-CCRI, μία από τις σημαντικότερες εταιρείες του τομέα. Στο άρθρο τους (ccri,2020) αναφέρουν την διαδικασία που ακολούθησαν για την οριοθέτηση των λιμένων. Η βάση της ανάλυσης τους είναι η λίστα των λιμανιών του World Port Index (World Port Index,2019). Με την χρήση AIS δεδομένων και των λιμανιών του World Port Index, δημιούργησαν συστάδες γύρω από τα σημεία στα οποία βρίσκονται τα λιμάνια. Ο αλγόριθμος που χρησιμοποιήθηκε για τον εντοπισμό συστάδων είναι ο DBSCAN , ένας αλγόριθμος ανάλυσης συστάδων βασισμένος στην πυκνότητα που ομαδοποιεί σημεία σε περιοχές μεγάλης πυκνότητας ενώ θεωρεί όλα τα σημεία σε περιοχές μικρής πυκνότητας εξαιρέσεις. Τέλος, υπολόγισαν το convex hull κάθε συστάδας για την οριοθέτηση των λιμένων. Επειδή δώσανε στον αλγόριθμο τα σημεία στα οποία θα δημιουργηθούν οι συστάδες, η προσέγγιση τους μπορεί να θεωρηθεί μια επιβεβλημένη παραλλαγή της συνηθισμένης ανάλυσης συστάδων.

Ένα άλλο παράδειγμα είναι η δημοσίευση των Cerdeiro *et al.* (2020). Στόχος τους ήταν να εκτιμήσουν τον όγκο του παγκοσμίου εμπορίου σε πραγματικό χρόνο με την χρήση ακατέργαστων δεδομένων AIS. Επειδή χρησιμοποίησαν ακατέργαστα δεδομένα, το μεγαλύτερο κομμάτι της εργασίας τους αφιερώθηκε στην οριοθέτηση των λιμένων και στην αναγνώριση των port calls. Για το φιλτράρισμα των δεδομένων χρησιμοποίησαν το πεδίο της ταχύτητας σε συνδυασμό με την κατάσταση πλοήγησης, με την υπόθεση ότι η κατάσταση πλοήγησης θα ενημερωθεί έγκαιρα και έγκυρα από το πλήρωμα τις περισσότερες φορές. Ο αλγόριθμος που χρησιμοποιήθηκε για τον εντοπισμό συστάδων λιμένων είναι μία weighted παραλλαγή του DBSCAN , ενώ παρόμοια, η οριοθέτηση των συστάδων έγινε με τον υπολογισμό του convex hull. Στην δημοσίευση τους αναφέρουν πως μια παρόμοια παραλλαγή ενός διαφορετικού αλγορίθμου, του OPTICS θα μπορούσε πιθανώς να βελτιώσει τα αποτελέσματα τους.

2.4 Μεθοδολογία

Για την επίλυση του προβλήματος θα κρατήσουμε και θα χτίσουμε πάνω στην μεθοδολογία που αναφέρθηκε στις σχετικές πηγές.

Το σημαντικότερο κομμάτι της ανάλυσης είναι ο εντοπισμός θαλάσσιων γεωγραφικών περιοχών ενδιαφέροντος βάσει των δεδομένων AIS. Αν και υπάρχει ήδη πηγή που καταγράφει τα σημαντικότερα λιμάνια του κόσμου και τα χαρακτηριστικά τους (World Port Index,2019), η δυνατότητα εντοπισμού λιμένων χωρίς την εισαγωγή βοηθητικών δεδομένων θα ήταν προτιμότερη καθώς επιτρέπει στον αλγόριθμο να ανταποκρίνεται καλύτερα σε αλλαγές όπως στην παύση λειτουργίας λιμένων ή στην κατασκευή καινούργιων. Για αυτόν τον λόγο, η μεθοδολογία των Cerdeiro *et al.* (2020) θεωρείται προτιμότερη.

Για το πρόβλημα του εντοπισμού θα χρησιμοποιηθεί η ανάλυση συστάδων όπως αναφέρθηκε. Πολλοί διαφορετικοί αλγόριθμοι είναι ικανοί να την υλοποιήσουν και μπορεί να διαφέρουν ως προς τον τρόπο λειτουργίας τους και τις ιδιότητές τους. Η αποτελεσματικότητα του κάθε αλγορίθμου εξαρτάται από το πρόβλημα και τους στόχους του αναλυτή. Συνήθως, η επιλογή του βέλτιστου αλγορίθμου και των παραμέτρων του είναι ένας συνδυασμός εμπειριστικών επιλογών και πειραματισμού και είναι μια από τις πιο χρονοβόρες διαδικασίες της ανάλυσης δεδομένων.

Στην δικιά μας εργασία, σκοπεύουμε να ομαδοποιήσουμε ένα σύνολο από γεωγραφικά στίγματα καραβιών βάσει της τοποθεσίας στην οποία βρίσκονται, έτσι ώστε τα στίγματα που είναι κοντά μεταξύ τους να είναι στην ίδια συστάδα. Επειδή κάθε συστάδα πρέπει να αντιπροσωπεύει μια γεωγραφική περιοχή ενδιαφέροντος όπως ένα λιμάνι ή ένα αγκυροβόλιο, θα πρέπει να συσταδοποιηθούν μόνο τα στίγματα που βρίσκονται μέσα σε τέτοιες περιοχές.

Αυτό σημαίνει πως οι βέλτιστες μεθοδολογίες για την επίλυση του προβλήματος θα είναι βασισμένες στην πυκνότητα και όχι στην εγγύτητα. Μέθοδοι βασισμένες στην πυκνότητα, είναι ικανές να δημιουργήσουν συστάδες με μη κυρτά ή πολύπλοκα σχήματα, όπως είναι τα λιμάνια, ενώ παράλληλα μπορούν να απομονώσουν εξαιρέσεις. Βέβαια, αξίζει να σημειωθεί πως δεν είναι όλες οι περιοχές μεγάλης κινητικότητας λιμάνια ή αγκυροβόλια. Ποτάμια και κανάλια μπορεί να έχουν αρκετά μεγάλη πυκνότητα στιγμάτων ώστε να δημιουργηθεί συστάδα. Για να το αποφύγουμε αυτό πρέπει να χρησιμοποιήσουμε και τα άλλα χαρακτηριστικά που παρέχονται στο μήνυμα AIS για να φιλτράρουμε τα δεδομένα. Μια καλή προσέγγιση είναι το φιλτράρισμα βάσει ταχύτητας καθώς τα καράβια που βρίσκονται μέσα στα λιμάνια τείνουν να έχουν πολύ χαμηλή ή μηδενική ταχύτητα. Άλλα πεδία όπως η κατάσταση πλοήγησης και ο τύπος πλοίου μπορούν επίσης να αξιοποιηθούν σε αυτό το στάδιο.

Ολοκληρώνοντας το κομμάτι του εντοπισμού θα πρέπει να γίνει κατάλληλη αξιολόγηση των αποτελεσμάτων πριν την οριοθέτηση τους. Πηγές που καταγράφουν λιμάνια όπως το World Port Index μπορούν να χρησιμοποιηθούν για αυτήν την αξιολόγηση.

Δεδομένου ότι το κομμάτι της ανάλυσης συστάδων έχει ολοκληρωθεί με ικανοποιητική ακρίβεια, επόμενο βήμα είναι η οριοθέτηση των θαλασσίων γεωγραφικών περιοχών που εντοπίστηκαν. Στόχος είναι, δεδομένου ενός συνόλου σημείων που ανήκουν σε κάποιο λιμάνι, να δημιουργηθεί ένα πολύγωνο αντιπροσωπευτικό αυτού του λιμανιού. Υπάρχουν δύο διαφορετικοί αλγόριθμοι που μπορούν να δώσουν αυτό το αποτέλεσμα, οι οποίοι αναλύονται στα επόμενο υποκεφάλαιο μαζί με τους αλγορίθμους ανάλυσης συστάδων.

2.5 Αλγόριθμοι

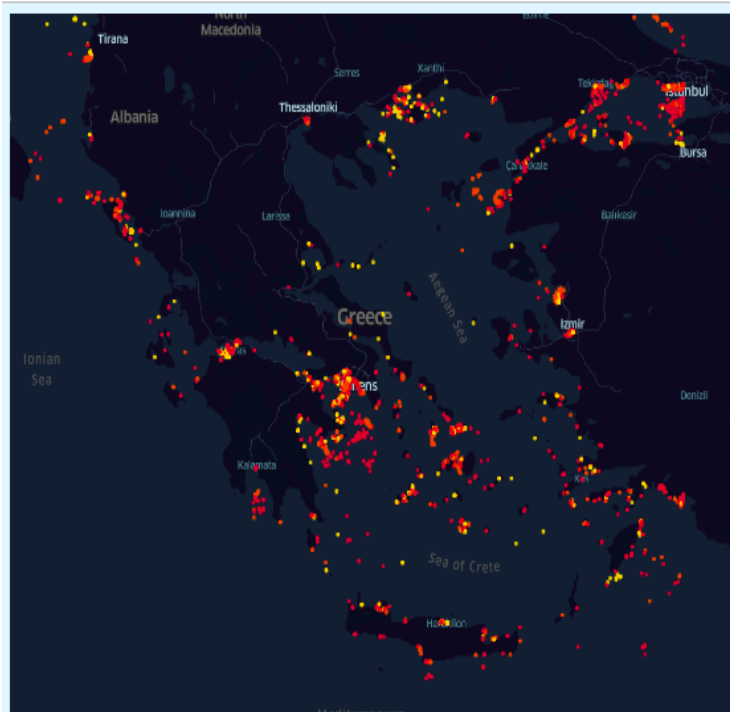
2.5.1 k-means clustering

Το προηγούμενο κεφάλαιο ανέφερε πως οι αλγόριθμοι συσταδοποίησης βασισμένοι σε πυκνότητα είναι η καλύτερη επιλογή για την επίλυση του προβλήματος. Ένας διαφορετικός αλγόριθμος που λάβαμε υπόψη είναι ο k-means. Ο k-means είναι ένας κλασικός αλγόριθμος συσταδοποίησης που έχει ως στόχο τον διαχωρισμό n παρατηρήσεων σε k συστάδες. Το k είναι μια παράμετρος που εισάγεται στον αλγόριθμο από τον αναλυτή, κάτι που σημαίνει πως ο αριθμός των συστάδων που θα δημιουργήσει ο αλγόριθμος θα πρέπει να είναι γνωστός πριν την εκτέλεση του. Αν και αυτός ο περιορισμός είναι σημαντικός, δεν κάνει από μόνος του τον αλγόριθμο ακατάλληλο, καθώς υπάρχουν τεχνικές για τον εντοπισμό της βέλτιστης τιμής του k που θα μπορούσαν να χρησιμοποιηθούν (Makles, 2012).

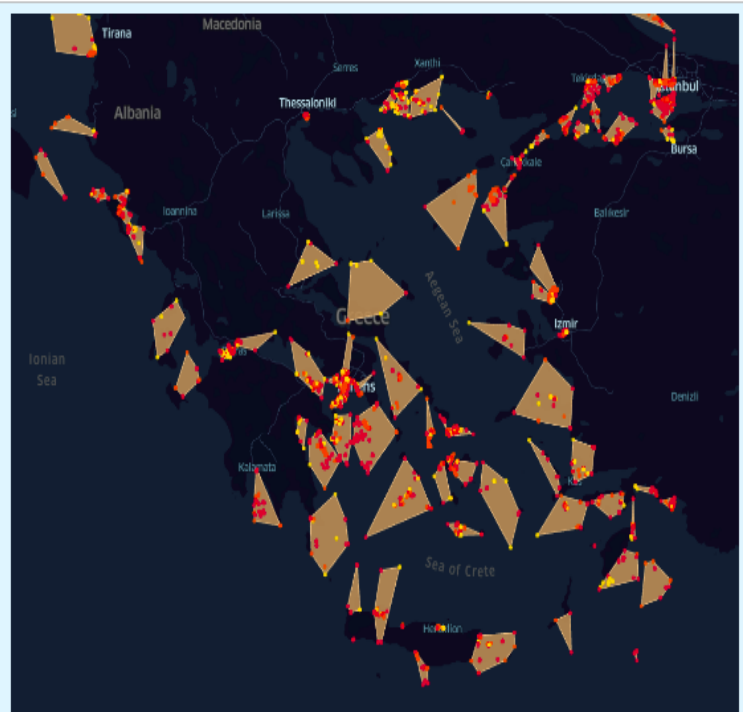
Ο μεγαλύτερος περιορισμός του k-means στην οριοθέτηση λιμένων είναι πως δεν επιτρέπει εξαιρέσεις στα δεδομένα. Ακόμα και αν φιλτραριστούν τα δεδομένα βάσει ταχύτητας, έτσι ώστε να μείνουν μόνο τα στίγματα των καραβιών που δεν βρίσκονται σε κίνηση, θα υπάρχουν στίγματα εκτός λιμανιών, είτε λόγω σφάλματος κατά την μετάδοσης είτε για κάποιο διαφορετικό λόγο. Αυτά τα στίγματα, αν προστεθούν σε μία συστάδα, θα επεκτείνουν τα όρια της και θα δημιουργήσουν προβλήματα στην διαδικασία της οριοθέτησης.

Αυτός ο περιορισμός παρουσιάζεται στο επόμενο παράδειγμα. Η αριστερή εικόνα δείχνει το σύνολο των στιγμάτων στην γενικότερη περιοχή της Ελλάδας με πολύ χαμηλή ταχύτητα. Για αυτά τα στίγματα, έγινε εκτέλεση του k-means και η οριοθέτηση των πολυγώνων που προέκυψαν. Τα αποτελέσματα φαίνονται στην δεξιά εικόνα:

Στίγματα AIS με ταχύτητα < 0.2 κόμβων στην περιοχή της Ελλάδας



Αποτέλεσμα k-means για k=100



Σίγουρα με ένα διαφορετικό k , τα αποτελέσματα του αλγορίθμου θα μπορούσαν να είναι καλύτερα, αλλά αυτό δεν λύνει το βασικότερο πρόβλημα του k -means. Επειδή δεν επιτρέπει εξαιρέσεις και ο υπολογισμός των πολυγώνων γίνεται βάσει των ακριανών σημείων της κάθε συστάδας, τα στίγματα που βρίσκονται μακριά από τα λιμάνια θα ορίζουν τις άκρες των πολυγώνων. Ακόμα και αν ο αλγόριθμος καταφέρει να συσταδοποιήσει σωστά το μεγαλύτερο κομμάτι των δεδομένων, ένα μόνο στίγμα που βρίσκεται μερικά χιλιόμετρα έξω από ένα λιμάνι και ανήκει στην συστάδα του είναι αρκετό για να επεκτείνει το πολύγωνο του κατά χιλιόμετρα. Επειδή είναι δύσκολο να αναγνωριστούν αυτές οι εξαιρέσεις στην προεπεξεργασία των δεδομένων είναι προτιμότερη η επιλογή ενός αλγορίθμου που μπορεί να τις χειριστεί καλύτερα.

2.5.2 DBSCAN

Ο DBSCAN (Density-based spatial clustering of applications with noise) (Ester *et al.* 1996) είναι ένας διαφορετικός αλγόριθμος ανάλυσης συστάδων με βάση την πυκνότητα. Στον DBSCAN, σημεία που βρίσκονται κοντά μεταξύ τους ομαδοποιούνται σε μια συστάδα ενώ σημεία σε περιοχές μικρής πυκνότητας μαρκάρονται ως εξαιρέσεις. Ο DBSCAN είναι αρκετά δημοφιλής καθώς σε αντίθεση με άλλους αλγόριθμους ανάλυσης συστάδων όπως ο k -means, ανταποκρίνεται πολύ καλύτερα στον θόρυβο και δεν απαιτεί από τον χρήστη γνώση για το πόσες συστάδες θα δημιουργηθούν πριν την εκτέλεση του.

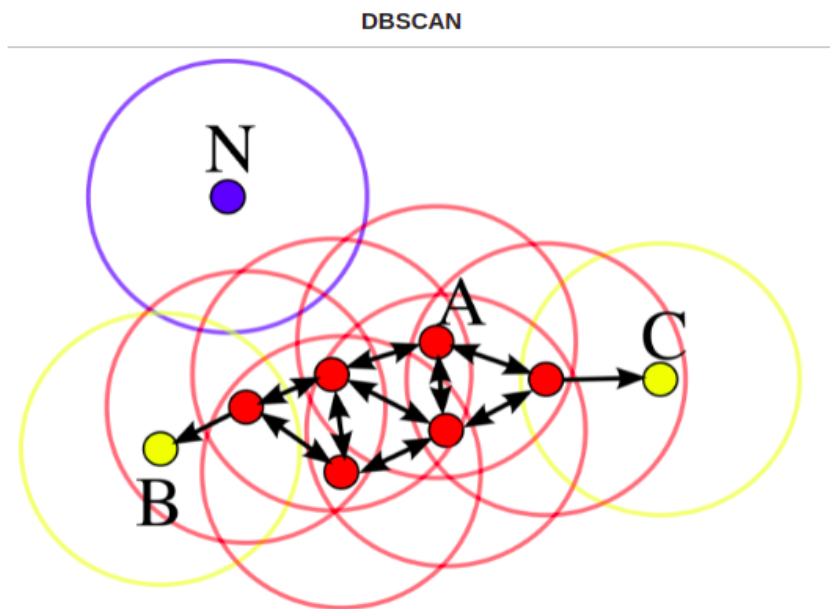
Χρειάζεται δύο παραμέτρους: Την ακτίνα ϵ που ορίζει το μέγεθος μιας περιοχής και έναν ακέραιο $minPts$ που ορίζει τον ελάχιστο αριθμό σημείων που πρέπει να έχει μια περιοχή για να θεωρηθεί αρκετά πυκνή για την δημιουργία συστάδας. Για γεωγραφικά στίγματα το ϵ πρέπει να αντιπροσωπεύει την γεωδαισιακή απόσταση - την συντομότερη απόσταση μεταξύ δύο σημείων στην επιφάνεια της γης.

Βάσει των παραμέτρων, ο DBSCAN κατηγοριοποιεί τα σημεία σε κεντρικά, άμεσα προσεγγίσιμα, πυκνά-προσεγγίσιμα και εξαιρέσεις ως εξής:

- Ένα σημείο p θεωρείται κεντρικό όταν υπάρχουν τουλάχιστον $minPts$ σημεία σε απόσταση ϵ από αυτό, συμπεριλαμβανομένου του p .
- Ένα σημείο q θεωρείται άμεσα προσεγγίσιμο από το p όταν βρίσκεται σε απόσταση ϵ από αυτό και το p είναι κεντρικό σημείο
- Ένα σημείο p είναι πυκνά-προσεγγίσιμο από ένα σημείο q , αν υπάρχει μονοπάτι $p_1, \dots, p_n$, με $p_1 = p$ και $p_n = q$, όπου κάθε p_{i+1} είναι άμεσα προσεγγίσιμο από το p_i , δηλαδή όλα τα σημεία του μονοπατιού πρέπει να είναι κεντρικά, με πιθανή εξαίρεση το q
- Όλα τα άλλα σημεία θεωρούνται εξαιρέσεις

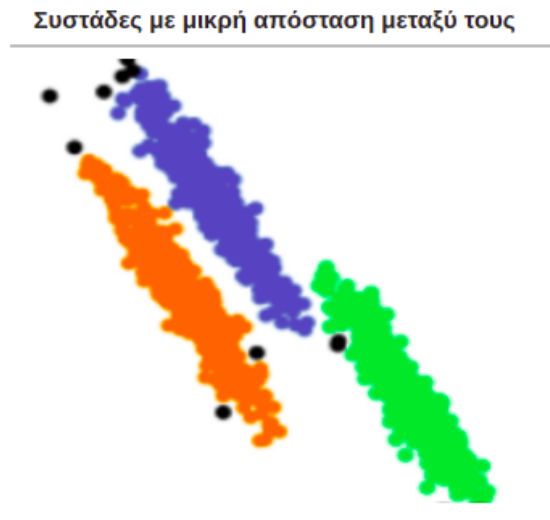
Για κάθε κεντρικό σημείο p , ο DBSCAN δημιουργεί μια συστάδα με το p και όλα τα σημεία που είναι πυκνά-προσεγγίσιμα από αυτό. Αυτό σημαίνει ότι κάθε συστάδα έχει τουλάχιστον ένα κεντρικό σημείο. Μη κεντρικά σημεία μπορεί να ανήκουν σε κάποια συστάδα καθώς δεν είναι όλα τα πυκνά-προσεγγίσιμα σημεία κεντρικά αλλά αντιπροσωπεύουν τα "όρια" της συστάδας.

Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα εκτέλεσης DBSCAN με $minPts=4$ και κάποια ακτίνα ϵ που φαίνεται από τον κύκλο που περιβάλλει κάθε σημείο. Όλα τα κόκκινα σημεία είναι κεντρικά καθώς υπάρχουν τουλάχιστον 4 σημεία μέσα στον κύκλο τους. Τα κίτρινα σημεία δεν είναι κεντρικά αλλά είναι και τα δύο τουλάχιστον πυκνά-προσεγγίσιμα από όλα τα κόκκινα κεντρικά σημεία. Βάσει του DBSCAN, τα κόκκινα και τα κίτρινα σημεία ανήκουν στην ίδια συστάδα, ενώ το μπλε σημείο αποτελεί εξαίρεση.



(<https://en.wikipedia.org/wiki/DBSCAN>)

Επειδή ο DBSCAN λειτουργεί με βάση την πυκνότητα, μπορεί να δημιουργήσει συστάδες με πολύπλοκα σχήματα, που δεν θα μπορούσαν να δημιουργηθούν εύκολα με διαφορετική κλάση αλγορίθμων. Στην κάτω εικόνα φαίνεται ένα τέτοιο παράδειγμα. Τα αποτελέσματα του DBSCAN δείχνουν πως μπορεί να διαχωρίσει ακόμα και συστάδες που βρίσκονται πολύ κοντά μεταξύ τους. Αυτό το χαρακτηριστικό του αλγορίθμου μπορεί να επιτρέψει τον εντοπισμό λιμανιών ακόμα και όταν η απόσταση μεταξύ τους είναι μικρή.



2.5.3 OPTICS

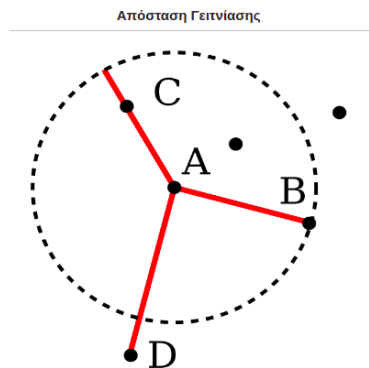
Παρά τα πλεονεκτήματα του DBSCAN, ένας από τους μεγαλύτερους περιορισμούς του είναι πως δεν επιτρέπει την δημιουργία συστάδων μεταβλητής πυκνότητας. Το ποια είναι η επιθυμητή πυκνότητα για την δημιουργία συστάδων ορίζεται από τον χρήστη και δεν μεταβάλλεται κατά την διαδικασία της εκτέλεσης του αλγορίθμου. Αυτό είναι ένα σημαντικό μειονέκτημα για τους στόχους της συγκεκριμένης εργασίας καθώς ο αριθμός καραβιών που εξυπηρετεί ένα λιμάνι μεταβάλλεται σημαντικά από λιμάνι σε λιμάνι. Για παράδειγμα, μπορεί μικρότερα νούμερα στις παραμέτρους του DBSCAN να εντοπίζουν με επιτυχία όλα τα λιμάνια του κόσμου, αλλά να δημιουργούν συστάδες σε σημεία που δεν θα έπρεπε (false positives). Αντίστοιχα, μπορεί μεγαλύτερα νούμερα στις παραμέτρους να δημιουργούν συστάδες μόνο σε σημεία που υπάρχουν λιμάνια, αλλά να μην εντοπίζουν με επιτυχία μικρότερα λιμάνια (false negatives). Αυτό το μειονέκτημα αντιμετωπίζει ο OPTICS.

Ο OPTICS(Ordering points to identify the clustering structure)(Ankerst *et al.*, 1999) είναι ένας αλγόριθμος συσταδοποίησης βασισμένος στην πυκνότητα και αναπτύχθηκε από κάποιους από τους αρχικούς συγγραφείς του DBSCAN ως μία επέκταση του αρχικού αλγορίθμου. Όπως και ο DBSCAN, ο OPTICS χρειάζεται δύο παραμέτρους: Τον ελάχιστο αριθμό σημείων που πρέπει να έχει μια περιοχή και κάποια ακτίνα ϵ . Η σημαντικότερη διαφορά είναι πως το ϵ δεν ορίζει το σταθερό μέγεθος μιας περιοχής όπως στον DBSCAN, αλλά το μέγιστο μέγεθος που μπορεί να έχει μια περιοχή. Αυτό κάνει τον OPTICS λιγότερο εξαρτημένο στις παραμέτρους του ενώ παράλληλα επιτρέπει την δημιουργία συστάδων διαφορετικού μεγέθους και πυκνότητας. Οι κατηγοριοποιήσεις των σημείων είναι οι ίδιες (κεντρικά, άμεσα προσεγγίσιμα, πυκνά-προσεγγίσιμα), αλλά προσθέτει δύο καινούργιους ορισμούς:

Κεντρική Απόσταση: Η ελάχιστη τιμή ε_{min} ($0 < \varepsilon_{min} \leq \varepsilon$) που χρειάζεται ένα σημείο για να θεωρηθεί ως κεντρικό. Κάθε σημείο έχει την δικιά του κεντρική απόσταση. Αν ένα σημείο δεν μπορεί να θεωρηθεί κεντρικό με ακτίνα ε , η κεντρική του απόσταση δεν ορίζεται.

Απόσταση Γειτνίασης: Για δύο κεντρικά σημεία p, q η απόσταση γειτνίασης τους ορίζεται ως η μεγαλύτερη τιμή ανάμεσα στην ευκλείδεια απόσταση του p και του q και την κεντρική απόσταση του p .

Στην παρακάτω εικόνα, για το κεντρικό σημείο A , φαίνεται η απόσταση γειτνίασης με τα σημεία B, C , και D με την κόκκινη γραμμή:

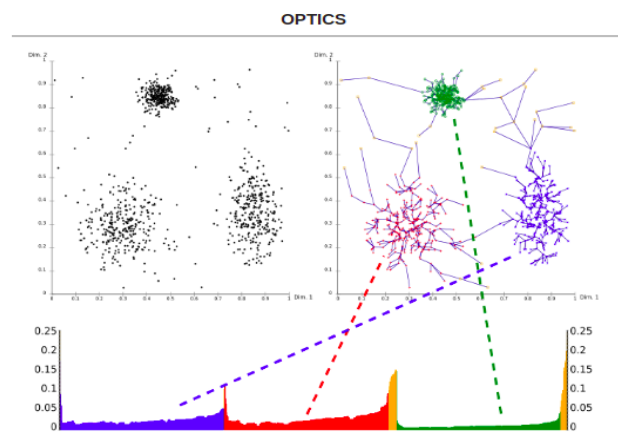


(https://en.wikipedia.org/wiki/Local_outlier_factor)

Ο OPTICS λειτουργεί ως εξής:

- Ξεκινώντας από ένα τυχαίο σημείο A , για κάθε σημείο που βρίσκεται στην γειτονιά του, υπολογίζει την απόσταση γειτνίασης από αυτό.
- Όλοι οι γείτονες εισάγονται σε μια ουρά προτεραιότητας βάσει της απόστασης γειτνίασης τους.
- Επιλέγεται το επόμενο σημείο βάσει της ουράς προτεραιότητας.

Με την ολοκλήρωση της εκτέλεσης, ο αλγόριθμος κατασκευάζει ένα διάγραμμα γειτνίασης όπου στον άξονα x βρίσκονται τα σημεία με την σειρά που επεξεργάστηκαν και στον άξονα y η χαμηλότερη απόσταση γειτνίασης που είχαν. Επειδή τα σημεία που ανήκουν σε μία συστάδα έχουν σχετικά χαμηλότερη απόσταση γειτνίασης με τον κοντινότερο τους γείτονα, οι συστάδες εμφανίζονται ως κοιλάδες στο διάγραμμα.

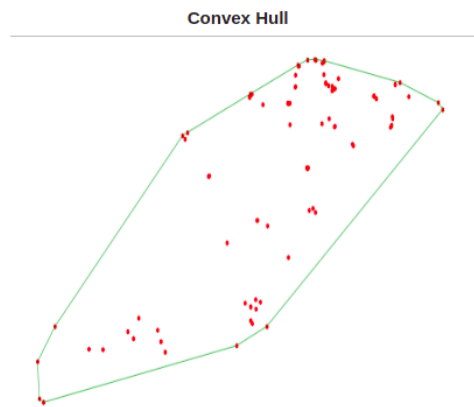


Η εξαγωγή των συστάδων από το διάγραμμα γειτνίασης τυπικά γίνεται με την χρήση ενός διαφορετικού αλγορίθμου για τον εντοπισμό των τοπικών ελαχίστων και μεγίστων. Επειδή ο OPTICS μπορεί να εντοπίσει συστάδες με διαφορετικά μεγέθη και πυκνότητα, θεωρούμε πως είναι ένας αλγόριθμος που αξίζει να ληφθεί υπόψη για την επίλυση του προβλήματος της οριοθέτησης θαλασσίων γεωγραφικών περιοχών.

2.5.4 Οριοθέτηση Στιγμάτων

Με την ολοκλήρωση της ανάλυσης συστάδων, έχουμε ως αποτέλεσμα ένα σύνολο στιγμάτων για κάθε λιμάνι που πρέπει να οριοθετηθούν. Στις σχετικές εργασίες αυτό επιτεύχθηκε με τον υπολογισμό του convex hull για κάθε συστάδα.

Το convex hull ορίζεται ως το μικρότερο δυνατό σχήμα που μπορεί να περικλείσει ένα σύνολο σημείων, έτσι ώστε για οποιαδήποτε δύο σημεία να μπορεί να υπάρξει ευθεία γραμμή που να τα ενώνει και να μην βγαίνει από τα όρια του σχήματος. Με άλλα λόγια, είναι το μικρότερο δυνατό κυρτό πολύγωνο που περιέχει όλα τα σημεία ενός συνόλου δεδομένων. Παράδειγμα ενός convex hull είναι το πολύγωνο στην παρακάτω εικόνα:



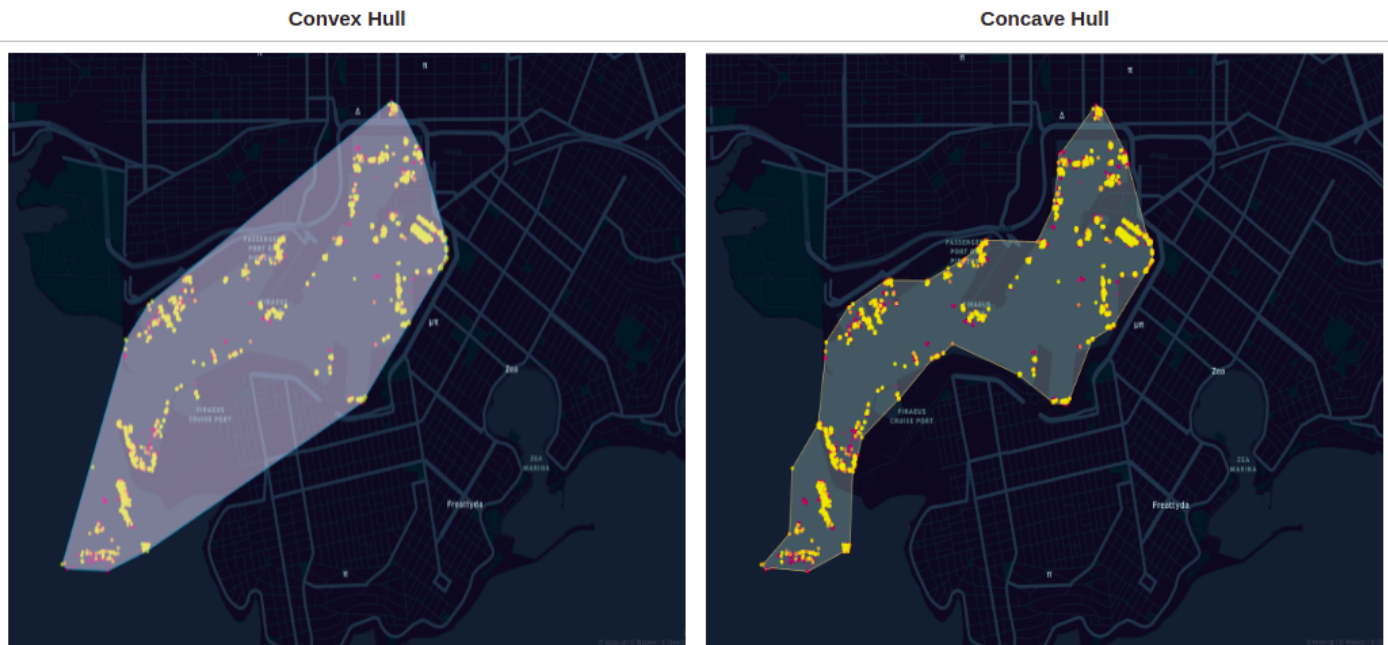
Το convex hull μπορεί να υπολογιστεί με σχετικά χαμηλή αλγοριθμική πολυπλοκότητα¹ και είναι μια αρκετά καλή προσέγγιση για τα όρια ενός λιμανιού. Αυτή η προσέγγιση μπορεί να βελτιωθεί αν ληφθούν υπόψη και μη-κυρτά σχήματα.

Για τον υπολογισμό ενός μη-κυρτού πιο πολύπλοκου πολυγώνου μπορεί να χρησιμοποιηθεί ο αλγόριθμος concave hull (Moreira & Santos, 2007). Αυτός ο αλγόριθμος αποτελεί μια εναλλακτική επιλογή για την οριοθέτηση σημείων και έχει ως στόχο τον υπολογισμό του καλύτερα δυνατού πολυγώνου που περιέχει όλα τα σημεία ενός πεπερασμένου συνόλου. Ο υπολογισμός του είναι πολυπλοκότερος² αλλά έχει βρει εφαρμογές στην ανάλυση συστάδων καθώς τείνει να διαχειρίζεται καλύτερα κάποια κοίλα σχήματα, κυρίως όταν τα στίγματα δεν είναι ίσα κατανεμημένα στον χώρο.

¹Επειδή η εκτέλεση του αλγορίθμου έγινε με την χρήση βιβλιοθήκης δεν είναι ξεκάθαρο ποιος αλγόριθμος χρησιμοποιείται για το convex hull και ποια είναι η πολυπλοκότητα του. Ένας από τους καλύτερους αλγορίθμους για τον υπολογισμό του convex hull είναι ο Chan's algorithm, που εκτελείται σε $O(n \log h)$ χρόνο, όπου h είναι αριθμός των κορυφών του convex hull.

² Στην αρχική δημοσίευση του αλγορίθμου δεν αναφέρεται η αλγοριθμική πολυπλοκότητα του. Ο ισχυρισμός ότι το concave hull είναι πολυπλοκότερο να υπολογιστεί βασίζεται στο ότι η εκτέλεση του αλγορίθμου είναι σημαντικά πιο αργή στην εκτέλεση του στην python.

Δεδομένου ενός συνόλου στιγμάτων, και οι δύο αλγόριθμοι μπορούν να θεωρηθούν κατάλληλοι για την οριοθέτηση τους. Στην παρακάτω εικόνα φαίνεται το σύνολο των γεωγραφικών στιγμάτων για το λιμάνι του Πειραιά, το μεγαλύτερο λιμάνι της Ελλάδας, και τα αποτελέσματα που έχει ο κάθε αλγόριθμος στην οριοθέτηση του¹:



Αν και το concave hull φαίνεται πιο αντιπροσωπευτικό του μεγέθους του λιμανιού στην παραπάνω περίπτωση, για το υπόλοιπο της εργασίας θα χρησιμοποιηθεί το convex hull για την οριοθέτηση των λιμένων. Ο βασικότερος λόγος για την επιλογή μας είναι ο όγκος των δεδομένων. Ακόμα και στα πλαίσια μιας μόνο χώρας, ο υπολογισμός του concave hull για κάθε λιμάνι είναι μια διαδικασία αρκετά απαιτητική σε υπολογιστικούς πόρους. Επίσης, το μόνο πλεονέκτημα του concave hull έναντι του convex hull είναι ότι μπορεί να μειώσει την περιοχή της στεριάς που περιλαμβάνει το πολύγωνο του λιμανιού. Αυτό το πλεονέκτημα είναι αμελητέο όταν στόχος της οριοθέτησης είναι να χρησιμοποιηθεί για να αναγνωριστεί αν ένα στίγμα βρίσκεται μέσα σε ένα λιμάνι ή όχι. Για αυτήν την περίπτωση, το κατά πόσο ένα πολύγωνο επεκτείνεται στην στεριά δεν έχει επιρροή στα στίγματα καθώς αυτά βρίσκονται στην θάλασσα.

¹Ο υπολογισμός του σχήματος που δημιουργείται από ένα σύνολο στοιχείων ονομάζεται α -shape. Το convex hull και το concave hull είναι και τα δύο τρόποι για να υπολογιστεί το α -shape των στιγμάτων. Τα μαθηματικά και ο τρόπος υπολογισμού τους είναι πέρα από τα όρια αυτής της εργασίας.

(κενή σελίδα)

3. Σχεδίαση

3.1 Προεπεξεργασία

Πριν το κομμάτι της υλοποίησης θα χρειαστεί η κατάλληλη προεπεξεργασία των δεδομένων. Το κομμάτι της προεπεξεργασίας θα γίνει με ξεχωριστή εκτέλεση προγράμματος. Αυτό είναι σημαντικό καθώς η ανάλυση δεδομένων είναι μια επαναλαμβανόμενη διαδικασία που πιθανώς χρειάζεται πολλές εκτελέσεις των αλγορίθμων με διαφορετικούς παραμέτρους. Η εκτέλεση της προεπεξεργασίας και η αποθήκευση των επεξεργασμένων δεδομένων πριν την χρήση τους επιτρέπει την αποδοτικότερη ανάλυση των δεδομένων.

Τα αρχικά δεδομένα βρίσκονται σε ημιδομημένη μορφή, αποθηκευμένα σε αρχεία json. Το κάθε αρχείο json περιλαμβάνει 10,000 στίγματα και υπάρχουν 13106 αρχεία με συνολικό μέγεθος 68GB. Ακολουθεί ένα παράδειγμα μεμονομένου μηνύματος AIS:

```
{"ship_and_cargo_type":52,"flag":"Cyprus","mmsi":209126000,"nmea":"!AIVDM,2,1,2,A,537L2L020Ht?A0K:20mDiA85ADN37J22222220I@B554000;j0DPSmD`53,0*7E\r\n!AIVDM,2,2,2,A,mhAiH4ai0C@,2*7C","length":28,"destination":"HARBOUR TOWAGE R'DAM","created_at":"2020-05-17T12:34:40.569071Z","flag_short_code":"CY","imo":8414154,"collection_type":"terrestrial","ais_version":0,"draught":4.7,"name":"MULTRATUG 16","width":10,"msg_type":5,"id":"659bdf7a-2460-5515-a4c3-75defc232c97","call_sign":"C4PF2","msg_id":"1589718873_209126000","timestamp":"2020-05-17T12:34:33Z","dimensions":{"A":10,"B":18,"C":5,"D":5}}
```

Το πρώτο και απαραίτητο κομμάτι της προεπεξεργασίας αφορά τα πεδία των δεδομένων που θα χρειαστούν για την ανάλυση. Όπως αναφέρθηκε στο πρώτο κεφάλαιο, δεν έχει κάθε μήνυμα AIS όλες τις πληροφορίες διαθέσιμες και τα πεδία που περιέχει το κάθε μήνυμα εξαρτώνται από τον τύπο του μηνύματος. Για την ανάλυση, τα σημαντικότερα πεδία είναι η ταχύτητα και η τοποθεσία με τα μηνύματα τύπου 1,2,3,18,19,27 να είναι τα μόνα που περιέχουν και τα δύο.

Καλή ιδέα θα ήταν να γίνει και εξέταση της χρησιμότητας κάποιων άλλων πεδίων που διαθέτει το AIS, συγκεκριμένα ο τύπος πλοίου και η κατάσταση πλοήγησης. Η κατάσταση πλοήγησης είναι ένα πεδίο που ενημερώνεται από το πλήρωμα και περιέχει την πληροφορία του πότε ένα καράβι είναι δεμένο (moored) σε ένα λιμάνι. Ο τύπος πλοίου μπορεί να χρησιμοποιηθεί για ένα δεύτερο σετ δεδομένων που να περιέχει μόνο τα εμπορικά πλοία και να γίνει προσπάθεια εντοπισμού μόνο των λιμανιών της Ελλάδας που εξυπηρετούν εμπορικά πλοία καθώς θα μπορούσε να καταλήξει σε χρήσιμη πληροφορία.

Ο τύπος πλοίου βρίσκεται στα μηνύματα τύπου 5,19,24 ενώ η κατάσταση πλοήγησης βρίσκεται στα μηνύματα τύπου 1,2,3,18,19. Η κατάσταση πλοήγησης είναι συνήθως διαθέσιμη στα μηνύματα που περιέχουν την τοποθεσία αλλά η χρήση του τύπου πλοίου είναι λίγο πιο δύσκολη. Πρόκειται όμως για μία σταθερή τιμή για κάθε καράβι, οπότε αν βρεθεί ένα μόνο μήνυμα που να την περιέχει μπορεί να προστεθεί και στο υπόλοιπο data set.

Το δεύτερο κομμάτι της προεπεξεργασίας είναι το φιλτράρισμα βάσει τοποθεσίας. Λόγω του όγκου των δεδομένων, η ανάλυση συστάδων θα υλοποιηθεί μόνο για τα λιμάνια της Ελλάδας, οπότε κάθε στίγμα που δεν προέρχεται από την γενικότερη περιοχή της Ελλάδας θα αφαιρεθεί.

Αυτό ολοκληρώνει το κομμάτι της προεπεξεργασίας. Αν και μπορεί να υπάρχουν δεδομένα με φαινομενικά έγκυρες, αλλά λάθος τιμές, ο εντοπισμός τους είναι δύσκολος και οι αλγόριθμοι που θα χρησιμοποιηθούν μπορούν να χειριστούν αρκετά καλά τον θόρυβο. Ως αποτέλεσμα, δεν θεωρείται πως θα χρειαστεί περαιτέρω προεπεξεργασία.

3.2 Γλώσσα Προγραμματισμού & Βιβλιοθήκες

Η υλοποίηση της ανάλυσης δεδομένων και του προγράμματος παρακολούθησης θα γίνει με την γλώσσα προγραμματισμού Python. Η python είναι μία από τις δημοφιλέστερες γλώσσες προγραμματισμού στον τομέα της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Ο λόγος για αυτό δεν είναι τυχαίος. Η python έχει ένα εξαιρετικό documentation και ένα από τα πληρέστερα toolset για την ανάπτυξη οποιασδήποτε εφαρμογής που χρησιμοποιεί μηχανική μάθηση. Αναφέρονται κάποιες από τις βασικές βιβλιοθήκες της python που βοήθησαν στην επιλογή της ως την γλώσσα υλοποίησης για την εργασία.

3.2.1 scikit-learn

Η scikit-learn ή sklearn είναι μία βιβλιοθήκη ανάλυσης δεδομένων της python που παρέχει υποστήριξη για την εκτέλεση αλγορίθμων μηχανικής μάθησης. Η βιβλιοθήκη της θα χρησιμοποιηθεί στην εργασία για την εκτέλεση του DBSCAN και του OPTICS. Ένα από τα πλεονεκτήματα της είναι ότι δεν βασίζεται στις βασικές υλοποιήσεις των αλγορίθμων, και παρέχει επιπλέον παραμέτρους για την τροποποίηση και την βελτιστοποίηση τους.

3.2.2 pandas

Η pandas είναι επίσης μία βιβλιοθήκη που χρησιμοποιείται συχνά στην ανάλυση δεδομένων. Ένα από τα σημαντικότερα κομμάτια της βιβλιοθήκης είναι η δομή δεδομένων που υποστηρίζει, το dataframe. Το dataframe είναι μια δομή που θυμίζει αρκετά έναν πίνακα μιας σχεσιακής βάσης δεδομένων καθώς αποθηκεύει τα δεδομένα σε γραμμές και στήλες, με την δυνατότητα αποθήκευσης διαφορετικού τύπου δεδομένων σε κάθε στήλη. Εκτός από το dataframe, η βιβλιοθήκη παρέχει εύκολους τρόπους για την προεπεξεργασία και εξερεύνηση των δεδομένων. Στην εργασία, το dataframe και η pandas θα χρησιμοποιηθούν ως η βασική δομή αποθήκευσης των δεδομένων AIS καθ όλη τη διάρκεια της προεπεξεργασίας και ανάλυσης τους.

3.2.3 geopandas

Η geopandas είναι μία επέκταση της pandas για γεωχωρικά δεδομένα. Χρησιμοποιεί την pandas και τους τύπους δεδομένων της και απλά προσθέτει λειτουργικότητα για διάφορες πράξεις πάνω σε γεωμετρίες. Η geopandas και η δομή της, το geodataframe, θα χρησιμοποιηθούν για την εγγραφή των δεδομένων σε geojson και για βασικές πράξεις πάνω στα πολύγωνα όπως τον υπολογισμό του convex hull τους.

3.2.4 matplotlib

Η matplotlib είναι μία βιβλιοθήκη για την δημιουργία διαγραμμάτων. Θα χρησιμοποιηθεί για την οπτικοποίηση μη γεωγραφικών δεδομένων όπως τα αποτελέσματα της ανάλυσης συστάδων.

Κάποιες άλλες βασικές βιβλιοθήκες που θα χρησιμοποιηθούν είναι οι numpy, shapely, redis-py, json και geojson.

3.3 Αναπαράσταση Δεδομένων

Για την αναπαράσταση των γεωγραφικών δεδομένων χρησιμοποιήθηκε το kepler.gl. Το kepler.gl είναι ένα ανοιχτού-κώδικα πρόγραμμα για ανάλυση και παρουσίαση γεωχωρικών δεδομένων μεγάλου όγκου. Τα εργαλεία που προσφέρει για την οπτικοποίηση και εξερεύνηση των δεδομένων θα είναι πολύτιμη βοήθεια για την αναγνώριση της σημαντικότητας των διαφορετικών πεδίων των δεδομένων AIS πάνω στο πρόβλημα του εντοπισμού λιμένων.

3.4 Βάση Δεδομένων

Για την αποθήκευση των πολυγώνων και των μηνυμάτων AIS, θα χρειαστεί η χρήση μιας βάσης δεδομένων. Βάση δεδομένων μπορεί να θεωρηθεί οποιοδήποτε σύστημα αποθήκευσης οργανωμένης πληροφορίας με τρόπο που κάνει την ανάκτηση της γρήγορη και εύκολη. Η επιλογή της κατάλληλης βάσης για ένα πληροφοριακό σύστημα είναι μια σημαντική επιλογή και εξαρτάται από πολλούς παράγοντες, όπως την μορφή των δεδομένων του συστήματος και τον όγκο τους. Όπως έχει αναφερθεί, ο όγκος των δεδομένων του AIS είναι δύσκολα διαχειρίσιμος. Μία βάση δεδομένων κατάλληλη για την εργασία πρέπει να έχει τα εξής χαρακτηριστικά:

- Να είναι ικανή να αποθηκεύσει αποδοτικά τα δεδομένα, καθώς και τα πολύγωνα που θα προκύψουν από την ανάλυση δεδομένων.
- Να επιτρέπει πράξεις με τις γεωμετρίες που έχει αποθηκευμένες, όπως query για τον εντοπισμό όλων των σημείων που βρίσκονται σε ένα πολύγωνο.
- Να παρέχει χωρικά indexes για την αποδοτικότερη ανάκτηση πληροφορίας.
- Να προσφέρει γενικότερη υποστήριξη για γεωχωρικά δεδομένα και εφαρμογές.
- Να είναι επεκτάσιμη ώστε να μπορεί να χειριστεί τον μεγάλο όγκο των δεδομένων.

3.4.1 PostGIS

Η πρώτη επιλογή που λάβαμε υπόψη για την αποθήκευση δεδομένων είναι η PostGIS, μία επέκταση της σχεσιακής βάσης δεδομένων PostgreSQL. Η PostGIS αναπτύχθηκε για να προσθέσει λειτουργικότητα και υποστήριξη για γεωχωρικά δεδομένα στην PostgreSQL. Καθώς είναι σχεδιασμένη ειδικά για γεωχωρικές εφαρμογές, πληρεί τις περισσότερες προϋποθέσεις μας. Παρέχει την δυνατότητα δημιουργίας χωρικών indexes και επιτρέπει την εκτέλεση πολύπλοκων queries βάσει των αποθηκευμένων γεωμετριών ενώ ταυτόχρονα παραμένει απλή και εύχρηστη.

Αν και η αρχική μορφή των δεδομένων AIS δεν είναι η πιο κατάλληλη για μία σχεσιακή βάση δεδομένων, όπως η PostGIS, ένα σύστημα παρακολούθησης πλοίων χρειάζεται μόνο κάποιες βασικές πληροφορίες όπως την τοποθεσία των πλοίων, την ταχύτητα τους και κάποια βασικά χαρακτηριστικά για να λειτουργήσει. Αυτά τα περιορισμένα δεδομένα είναι δομημένα και μπορούν να αποθηκευτούν ευκολότερα σε κάποιον πίνακα στην βάση δεδομένων.

Ένας περιορισμός της PostGIS είναι στην επεκτασιμότητα της. Ένα πληροφοριακό σύστημα που κρατάει ιστορικά δεδομένα για τα περισσότερα καράβια του κόσμου μπορεί εύκολα να φτάσει τα terabyte δεδομένων. Οι κλασικές σχεσιακές βάσεις δεδομένων όπως η PostGIS έχουν χαμηλή απόδοση στην ανάκτηση μεγάλου όγκου δεδομένων και πολλές φορές δεν είναι η καλύτερη επιλογή για την διαχείρισή τους.

Γενικότερα, θεωρούμε πως η PostGIS είναι μια καλή και εύχρηστη επιλογή και μπορεί να χρησιμοποιηθεί για την αποθήκευση των επεξεργασμένων δεδομένων AIS για τους στόχους της εργασίας. Παρ'όλα αυτά δεν επιλέχθηκε καθώς θεωρήθηκε πως υπάρχουν καλύτερες εναλλακτικές επιλογές για την υλοποίηση του συστήματος.

3.4.2 MongoDB

Μία εναλλακτική επιλογή για την αποθήκευση δεδομένων είναι η mongoDB, μία NoSQL (Not only SQL) βάση δεδομένων. Οι NoSQL βάσεις έχουν βρει σημαντική χρήση τα τελευταία χρόνια σε εφαρμογές big data καθώς μπορούν να χειριστούν αποδοτικότερα τα μεγάλου όγκου δεδομένα. Ένας τρόπος με τον οποίο το επιτυγχάνουν αυτό είναι η απλούστερη μορφή αποθήκευσης των δεδομένων τους. Τα δεδομένα σε μία NoSQL βάση δεδομένων αποθηκεύονται σε ημιδομημένη ή αδόμητη μορφή χωρίς πολύπλοκες σχέσεις, κάτι που επιτρέπει τον διαχωρισμό τους σε πολλούς διαφορετικούς υπολογιστικούς κόμβους. Η χρήση περισσότερων υπολογιστικών κόμβων για την βελτίωση της απόδοσης ενός πληροφοριακού συστήματος είναι γνωστή και ως οριζόντια κλιμάκωση.

Η mongoDB είναι μία από τις δημοφιλέστερες NoSQL βάσεις δεδομένων και έχει το πλεονέκτημα ότι προσφέρει κάποια βασική υποστήριξη για γεωχωρικά δεδομένα. Πιο συγκεκριμένα, επιτρέπει την δημιουργία χωρικών indexes για την αποδοτικότερη χρήση των δεδομένων και την εκτέλεση βασικών γεωχωρικών queries για τις γεωμετρίες που έχει αποθηκευμένες.

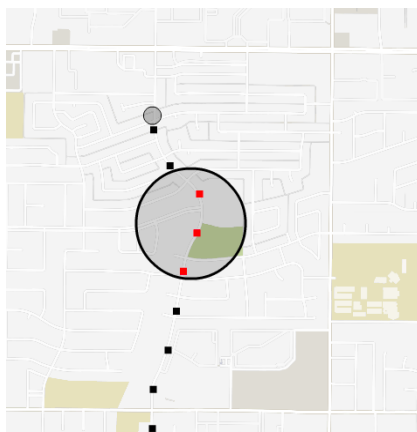
Συγκριτικά με την PostGIS, η mongoDB προσφέρει μεγαλύτερη επεκτασιμότητα ενώ ταυτόχρονα μπορεί να αποθηκεύσει τα AIS δεδομένα στην αρχική μορφή τους καθώς χρησιμοποιεί την μορφή JSON για την αποθήκευση των δεδομένων της. Βέβαια, αν και η mongoDB μπορεί να υποστηρίξει γεωχωρικές εφαρμογές, δεν είναι εξειδικευμένη για αυτές. Ως αποτέλεσμα, θυσιάζει στο κομμάτι της ανάλυσης των γεωχωρικών δεδομένων και οι συναρτήσεις και τα queries που επιτρέπει στις γεωμετρίες της είναι πιο περιορισμένα.

3.4.3 Tile38

Το tile38 είναι μία δομή αποθήκευσης γεωχωρικών δεδομένων, που μπορεί να χρησιμοποιηθεί ως NoSQL βάση δεδομένων. Βασίζεται στο redis και ουσιαστικά αποτελεί μία επέκταση του για γεωχωρικά δεδομένα. Η μορφή των αποθηκευμένων δεδομένων του tile38 όπως και του redis είναι κλειδιά-τιμές. Αυτή η μορφή δεδομένων επιτρέπει την οριζόντια κλιμάκωση της βάσης με την χρήση περισσότερων υπολογιστικών κόμβων, παρόμοια με την MongoDB. Σαν γεωχωρική βάση δεδομένων υποστηρίζει γεωχωρικά indexes για τον γρήγορο υπολογισμό γεωμετρικών πράξεων αλλά περιορίζεται στα queries που επιτρέπει.

Το σημαντικότερο πλεονέκτημα του tile38 έναντι των άλλων επιλογών είναι η υποστήριξη που παρέχει για το geofencing των αποθηκευμένων πολυγώνων. Geofencing ονομάζεται η τεχνική δημιουργίας μιας εικονικής περιμέτρου σε κάποιο γεωγραφικό σημείο με στόχο τον εντοπισμό γεγονότων που αφορούν την περίμετρο όπως όταν ένα αντικείμενο εισέρχεται στην περιοχή ή βγαίνει από αυτήν.

Παράδειγμα geofencing από την σελίδα του tile38:



Το geofencing του tile38 ακολουθεί το μοντέλο pubsub (publish-subscribe) όπου τα προγράμματα κατηγοριοποιούνται σε publishers και subscribers. Στο tile38, ένας publisher παρακολουθεί κάποιο πολύγωνο για συγκεκριμένα γεγονότα εντοπισμού. Τα γεγονότα μετά δημοσιεύονται σε ένα κανάλι σε μία συγκεκριμένη κατηγορία. Στο μοντέλο pubsub οι publishers δεν γνωρίζουν ποιοι είναι οι αποδέκτες των μηνυμάτων τους. Οι αποδέκτες εγγράφονται σε κατηγορίες και κανάλια, όχι σε συγκεκριμένους publisher, και λαμβάνουν όλα τα μηνύματα που δημιουργούνται από τις κατηγορίες που παρακολουθούν.

Επειδή το tile38 μπορεί να καλύψει τις ανάγκες της εργασίας και παρέχει εγγενή υποστήριξη για τον εντοπισμό των γεγονότων των καραβιών, ενώ ταυτόχρονα επιτρέπει την μελλοντική επέκταση του συστήματος, θεωρήθηκε πως είναι η καλύτερη επιλογή για την υλοποίηση.

Εδώ αξίζει να σημειωθεί πως η επιλογή μας έγινε καθαρά λόγω της χρησιμότητας του tile38 για τους στόχους της εργασίας και γενικότερα δεν θεωρούμε πως το redis και κατ' επέκταση το tile38 είναι η καλύτερη επιλογή για την επίμονη αποθήκευση των AIS δεδομένων όταν στόχος δεν είναι το geofencing. Για ένα διαφορετικό πληροφοριακό σύστημα, μία τεχνολογία όπως η HBase 2(Liu,2017) μπορεί να δώσει καλύτερα αποτελέσματα στην αποθήκευση δεδομένων μεγάλου όγκου που απαιτούν πολλές εγγραφές όπως τα δεδομένα AIS.

4. Υλοποίηση Ανάλυσης Συστάδων

4.1 Benchmarking

Αρχικά, χρειάστηκε ένα σημείο αναφοράς για τα αποτελέσματα των αλγορίθμων ώστε να γίνει η κατάλληλη επιλογή των παραμέτρων τους. Η αναπαράσταση των πολυγώνων στον χάρτη δεν είναι αρκετή για την αξιολόγηση των αποτελεσμάτων τους αν δεν είναι γνωστές οι τοποθεσίες των λιμανιών. Στις σχετικές εργασίες χρησιμοποιήθηκε το World Port Index, μία από τις πληρέστερες πηγές λιμανιών στον κόσμο. Το world Port Index καταγράφει 67 λιμάνια για την Ελλάδα μαζί με πληροφορίες όπως την τοποθεσία τους, το μέγεθος τους, τα καράβια που εξυπηρετεί και άλλα. Ένας περιορισμός του είναι ότι δεν καταγράφει πολλά από τα μικρότερα λιμάνια που υπάρχουν.

Μία διαφορετική πηγή που βρήκαμε είναι το [World Port Source](#). Σε παγκόσμια κλίμακα περιλαμβάνει λιγότερα λιμάνια από το World Port Index, αλλά στην περιοχή της Ελλάδας καταγράφει 103 λιμάνια και μπορεί να θεωρηθεί μία πιο ολοκληρωμένη πηγή για τα μικρότερα λιμάνια της Ελλάδας. Σαν αποτέλεσμα, επιλέχθηκε το World Port Source σαν σημείο αναφοράς για τα πολύγωνα των αλγορίθμων.

Ένα πρόβλημα της πηγής είναι ότι, από όσο γνωρίζουμε, το σετ δεδομένων τους δεν είναι διαθέσιμο σε ένα μεμονωμένο αρχείο, και οι πληροφορίες των λιμανιών παρουσιάζονται σε διαφορετική σελίδα για κάθε λιμάνι. Για την δημιουργία του σετ δεδομένων των λιμανιών έγινε η υλοποίηση ενός προγράμματος web scraping με την βιβλιοθήκη beautifulsoup της python. Παρακάτω φαίνεται η σελίδα του World Port Source:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z			Port of Corfu	
Most Popular			Contact Information	
ioi Theodoroi Port of Agios Theodoroi	Corfu Port of Corfu	Thessaloniki Port of Thessaloniki *	Port Location: Corfu	
xandroupolis Port of Alexandroupolis	Piraeus Port of Piraeus *	Volos Port of Volos *	Port Name: Port of Corfu	
A			Local Port Name: Port of Kerkira	
hladi Port of Achladi	Alexandroupolis Port of Alexandroupolis	Argostoli Port of Argostoli	Port Authority: Port Authority of Corfu	
amas Port of Adamas	Aliverion Port of Aliverion	Aspropyrgos Port of Aspropyrgos	Address: Xen Stratigou 2 Corfu Greece	
gina Port of Aegina	Alyki Alyki Harbor	Astakos Port of Astakos	Phone: 30 266 103 2655	
hia Marina Port of Aghia Marina	Amfilochia Port of Amfilochia	Avra Avra Aghia Marina	Fax: 30 266 103 9918	
hia Trias Olea Oil Terminal	Antikyra Port of Antikyra	Ayios Nikolaos Port of Ayios Nikolaos	800 Number:	
ioi Theodoroi Port of Agios Theodoroi	Antirio Port of Antirio		Email:	
C			Web Site:	
alkis Port of Chalkis	Chios Port of Chios	Corfu Port of Corfu	Latitude: 39° 37' 45" N	
ania Port of Chania	Corfu Gouvia Marina	Corinth Corinth Canal	Longitude: 19° 54' 44" E	
D			UNICODE: GRCFU	
			Port Type: Harbor	
			Port Size: Small	
			Port of Corfu on Social Media	

Ο μόνος τρόπος για να συλλεχθούν όλες οι πληροφορίες για όλα τα λιμάνια της Ελλάδας είναι να περάσει ο κώδικας από την κάθε μεμονωμένη σελίδα που υπάρχει για κάθε λιμάνι. Το index της σελίδας για τα ελληνικά λιμάνια περιέχει όλα τα url για τα ελληνικά λιμάνια. Με την κλάση SoupStrainer μπορεί να γίνει επιλογή συγκεκριμένων html tags. Ο παρακάτω κώδικας χρησιμοποιήθηκε για μία πρώτη εξαγωγή όλων των url από την σελίδα:

```
import requests
from bs4 import BeautifulSoup, SoupStrainer

page = requests.get("http://www.worldportsource.com/ports/index/GRC.php")
soup = BeautifulSoup(page.content, "html.parser", parse_only=SoupStrainer("a", href=True))
for a in soup:
    print(a['href'])
```

```
(base) ntkos@ntkos-IdeaPad-Gaming-3-15ARH05:~/Desktop$ python3
/ports/GRC_Port_of_Samos_2031.php
/ports/GRC_Port_of_Seriphos_3439.php
/ports/GRC_Port_of_Skala_Oropou_4614.php
/ports/GRC_Port_of_Skaramanga_2043.php
/ports/GRC_Port_of_Souda_2012.php
/ports/GRC_Port_of_Stratoni_3441.php
/ports/GRC_Port_of_Stylis_2032.php
/ports/GRC_Syros_Harbour_2035.php
#top
/ports/review/GRC_Port_of_Thessaloniki_1
/ports/GRC_Port_of_Tinos_3440.php
#top
/ports/review/GRC_Port_of_Volos_1042.php
/ports/GRC_Voudhia_Harbor_4602.php
#top
/ports/GRC_Port_of_Yali_2033.php
/ports/GRC_Port_of_Yerakini_358.php
#top
/ports/GRC_Port_of_Zante_2034.php
/countries.php
/states.php
/ports/index/GRC.php
/ports/GRC.php
/shipping/country/ports/GRC.php
/trade/GRC.php
#A
#C
#D
#E
#G
#I
#K
#L
#M
#N
#P
#R
#S
#T
#V
#Y
#Z
/ports/GRC_Port_of_Agioi_Theodoroi_2036.php
/ports/GRC_Port_of_Alexandroupolis_3315.php
/ports/GRC_Port_of_Corfu_2014.php
```

Από τα αποτελέσματα και μία επόμενη επίσκεψη στην ιστοσελίδα προέκυψαν οι εξής πληροφορίες:

- Όλα τα url των λιμανιών έχουν την μορφή 'ports/GRC_' ή 'ports/review/GRC_'.
- Οι 'ports/review/GRC_' σελίδες περιέχουν μόνο ιστορικές πληροφορίες για ένα λιμάνι και δεν είναι χρήσιμες για την ανάλυση δεδομένων.
- Για κάθε σελίδα της μορφής 'ports/review/GRC_' υπάρχει και σελίδα της μορφής 'ports/GRC_' με τις βασικές πληροφορίες για το λιμάνι.
- Για τις σελίδες που έχουν ιστορικές πληροφορίες, τα url των σελίδων που οδηγούν στις βασικές πληροφορίες τους δεν είναι διαθέσιμα στο index.
- Κάποια url εμφανίζονται δύο φορές.

Ο παρακάτω κώδικας εξαγει την ολοκληρωμένη λίστα με τα url των λιμανιών, και συλλέγει τις πληροφορίες από κάθε λιμάνι σε ένα pandas dataframe.

```
import requests
from bs4 import BeautifulSoup, SoupStrainer
import pandas as pd

page = requests.get("http://www.worldportsource.com/ports/index/GRC.php")
soup = BeautifulSoup(page.content, "html.parser", parse_only=SoupStrainer("a", href=True))

l = []
for a in soup:
    url = a['href']
    if url.startswith("/ports/GRC_"):
        l.append(url)
    elif url.startswith("/ports/review/GRC_"):
        l.append(url.replace("review/", ""))

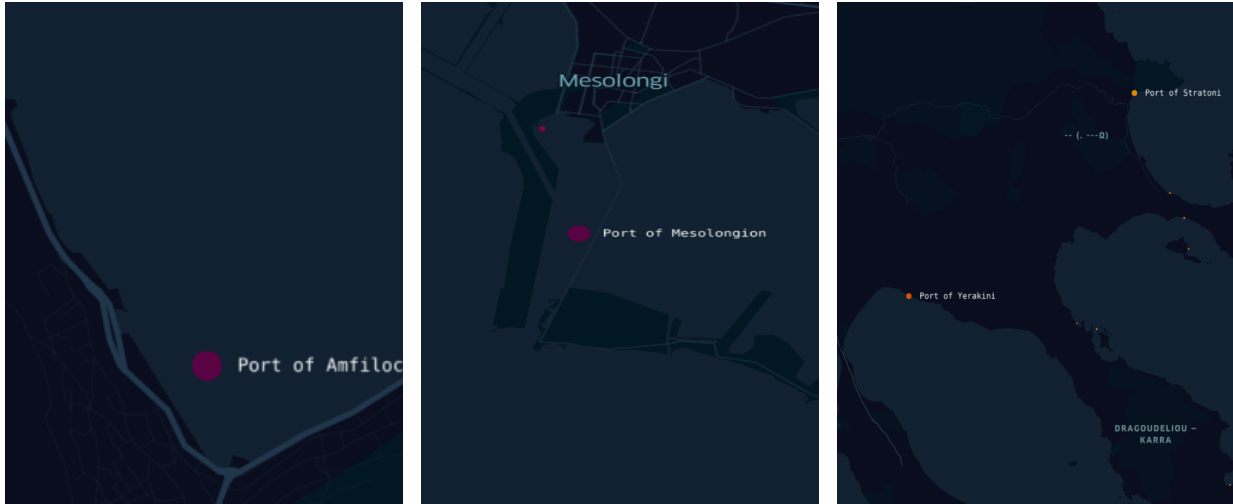
l = set(l) #remove duplicates
ports_list = []
for port in l:
    page = requests.get('http://www.worldportsource.com'+port)
    soup = BeautifulSoup(page.content, "html.parser", parse_only=SoupStrainer('tr'))
    port_data = []
    for row in soup:
        rows = row.text.strip().split("\n")
        title = rows[0]
        if title in ('Port Name:', 'Latitude:', 'Longitude:', 'Port Size:'):
            port_data.append(rows[2])

    ports_list.append(port_data)

df = pd.DataFrame(columns=['port', 'latitude', 'longitude', 'size'], data=ports_list)
```

Με την ολοκλήρωση της συλλογής των δεδομένων των λιμανιών, έγινε η μετατροπή των γεωγραφικών συντεταγμένων σε δεκαδικό σύστημα και αποθηκεύτηκαν σε ένα εξωτερικό αρχείο.

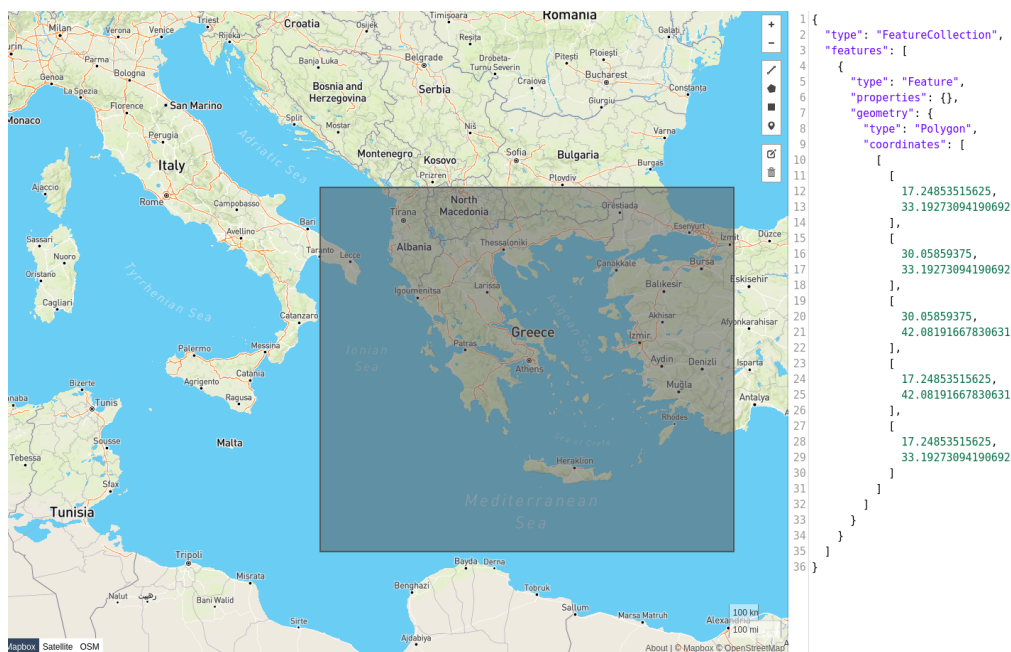
Από τα λιμάνια του world port source προκύπτει και ένας σημαντικός περιορισμός της ανάλυσης δεδομένων της εργασίας. Υπάρχουν λιμάνια τα οποία δεν έχουν κανένα στίγμα AIS κοντά τους, όπως φαίνεται και στις παρακάτω εικόνες. Αυτό σημαίνει πως ανεξάρτητα με την μεθοδολογία της εργασίας, ο εντοπισμός όλων των ελληνικών λιμανιών με την χρήση των δεδομένων του αυτόματου συστήματος αναγνώρισεως είναι αδύνατος. Ο λόγος που εμφανίζονται ελάχιστα ή καθόλου στίγματα σε κάποια μικρότερα λιμάνια δεν είναι γνωστός σε εμάς. Μία υπόθεση είναι ότι συσχετίζεται με την νομοθεσία περί AIS, που υποχρεώνει μόνο κάποια καράβια να διαθέτουν πομπούς AIS αλλά ίσως να οφείλεται και σε σφάλμα των δεδομένων που χρησιμοποιήθηκαν.



4.2 Προεπεξεργασία Δεδομένων

Για κάθε μήνυμα έγινε φιλτράρισμα βάσει της ταχύτητας και της τοποθεσίας των στιγμάτων ώστε να παραμείνουν μόνο τα στίγματα με ταχύτητα μικρότερη των 0.2 κόμβων και τοποθεσία την γενικότερη περιοχή της Ελλάδας. Επιλέχθηκε αυτή η ταχύτητα έναντι των μηδέν κόμβων καθώς το σύστημα AIS μερικές φορές καταγράφει πολύ χαμηλές τιμές, διάφορες του μηδέν σε στίγματα που βρίσκονται μέσα σε λιμάνια.

Για το φιλτράρισμα της τοποθεσίας δημιουργήθηκε χειροκίνητα ένα πολύγωνο για την γενικότερη περιοχή της Ελλάδας στο geojson.io όπως φαίνεται στην εικόνα:



Οι γεωγραφικές συντεταγμένες του πολυγώνου στρογγυλοποιήθηκαν στα τρία δεκαδικά και κρατήθηκαν μόνο τα σημεία τα οποία βρίσκονται μέσα στο πολύγωνο.

Ακολουθεί ένα δείγμα του κώδικα της προεπεξεργασίας, όπου οι μεταβλητές minlat,maxlat,minlon,maxlon είναι τα όρια του πολυγώνου:

```
cols = ['mmsi', 'latitude', 'speed', 'msg_type', 'timestamp',  
        'longitude', 'status', 'ship_and_cargo_type', 'ship_type', 'destination', 'eta']  
msg_types=[1,2,3,4,5,18,19,24,27] #mmsi,latitude,longitude,speed  
maindf = pd.DataFrame(columns=cols)  
for f in os.listdir(folder):  
    df = pd.read_json(folder+'/'+f)  
    df = df[df.speed<0.2]  
    df = df[df.msg_type.isin(msg_types)]  
    df = df[(df.latitude>minlat) & (df.latitude<maxlat) & (df.longitude>minlon) & (df.longitude<maxlon)]
```

Για την προσθήκη του τύπου πλοίου στο σετ των δεδομένων έγινε η εξαγωγή όλων των μοναδικών αναγνωριστικών των πλοίων και αναζήτηση τους στα αρχικά δεδομένα. Έπειτα δημιουργήθηκε ένα αρχείο με την αντιστοίχιση του κάθε αναγνωριστικού πλοίου με τον τύπο του, ο οποίος αργότερα προστέθηκε στα δεδομένα της ανάλυσης.

Το τελευταίο κομμάτι της προεπεξεργασίας αφορά την προετοιμασία των δεδομένων για την χρήση τους από τους αλγορίθμους συσταδοποίησης DBSCAN και OPTICS καθώς, λόγω των περιορισμένων υπολογιστικών πόρων, η εκτέλεση των αλγορίθμων με την χρήση όλων των δεδομένων είναι αδύνατη.

Για τον DBSCAN, επιλέχθηκε μία weighted παραλλαγή του για την μείωση του όγκου των δεδομένων, όπως έγινε στην εργασία των Cerdeiro *et al.* (2020). Αυτή η παραλλαγή είναι διαθέσιμη στην βιβλιοθήκη scikit-learn της python. Για την προετοιμασία των δεδομένων για τον Weighted DBSCAN έγινε στρογγυλοποίηση της τοποθεσίας των στιγμάτων AIS στα τρία δεκαδικά. Έπειτα, αφαιρέθηκαν τα διπλότυπα δεδομένα και προστέθηκε μία καινούργια στήλη για το βάρος του κάθε στίγματος. Το βάρος αντιπροσωπεύει τον αριθμό των στιγμάτων που βρίσκονται σε αυτήν την τοποθεσία.

Αν και η αφαίρεση του τέταρτου δεκαδικού στο γεωγραφικό πλάτος και μήκος θυσιάζει μέρος της ακρίβειας των δεδομένων, το σφάλμα στην αναπαράσταση του στίγματος δεν μπορεί να είναι μεγαλύτερο των εκατό μέτρων. Για τον εντοπισμό του λιμανιού στο οποίο βρίσκεται το στίγμα ενός καραβιού, αυτή η απόσταση θεωρείται αμελητέα. Για να επιβεβαιώσουμε την υπόθεση, δοκιμάστηκε και στρογγυλοποίηση των δεδομένων στα τέσσερα δεκαδικά χωρίς διαφορετικά αποτελέσματα.

Η παραπάνω τεχνική μπορεί να χρησιμοποιηθεί μόνο για τον DBSCAN. Από όσο γνωρίζουμε, δεν υπάρχει παραλλαγή του OPTICS που να επιτρέπει την εκτέλεση weighted δεδομένων. Για την εκτέλεση του OPTICS έγινε τυχαία δειγματοληψία (1%) στο αρχικό σετ δεδομένων με σταθερό seed για την δυνατότητα αναπαραγωγής των αποτελεσμάτων.

Με την ολοκλήρωση της προεπεξεργασίας, καταφέραμε να μειώσουμε τον όγκο των δεδομένων για την Ελλάδα σε μέγεθος κατάλληλο για την χρήση τους από τους αλγορίθμους ανάλυσης συστάδων.

Συνολικά δημιουργήθηκαν τα εξής datasets:

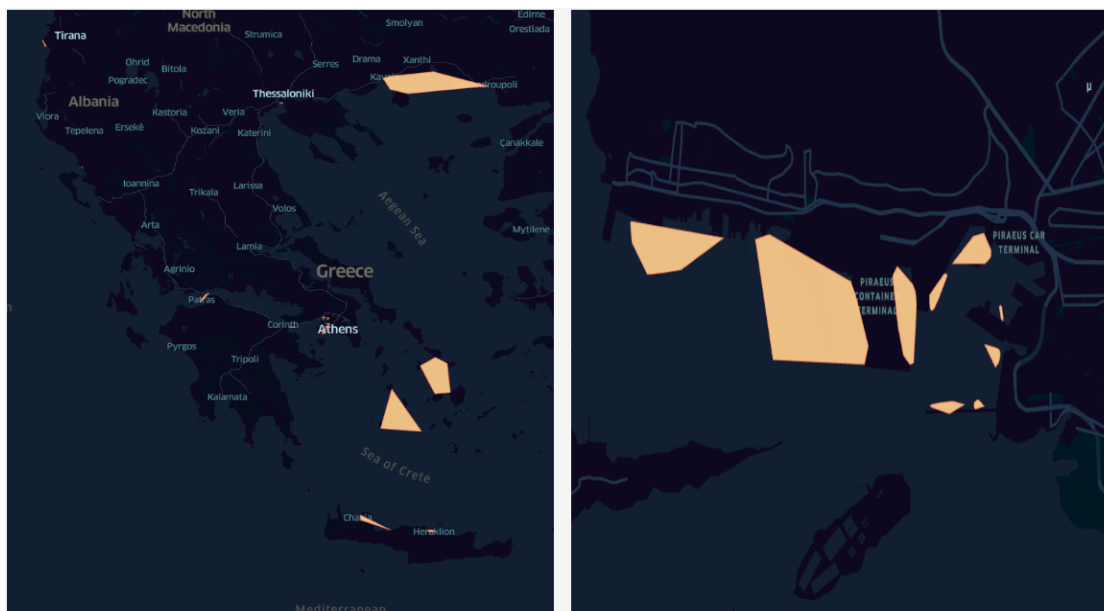
- Στίγματα στην Ελλάδα με ταχύτητα μικρότερη των 0.2 κόμβων.
- Εμπορικά πλοία στην Ελλάδα με ταχύτητα μικρότερη των 0.2 κόμβων.
- Στίγματα στην Ελλάδα με ταχύτητα μικρότερη των 0.2 κόμβων και κατάσταση πλοήγησης: moored
- Εμπορικά πλοία στην Ελλάδα με ταχύτητα μικρότερη των 0.2 κόμβων και κατάσταση πλοήγησης: moored

Ο όγκος των δεδομένων μειώθηκε από τα ~65GB στα ~200MB για το μεγαλύτερο dataset, τα στίγματα της Ελλάδας με χαμηλή ταχύτητα.

4.3 Εκτέλεση OPTICS

Ο OPTICS είναι ο πρώτος αλγόριθμος ανάλυσης συστάδων που δοκιμάστηκε. Τα αποτελέσματα μιας απλής εκτέλεσης του OPTICS φαίνονται στις παρακάτω εικόνες:

OPTICS $maxEps=infinity$



Όπως φαίνεται, για περιοχές μικρότερης πυκνότητας όπως είναι οι κυκλάδες, ο αλγόριθμος δημιουργεί τεράστιες συστάδες που καλύπτουν πολλά λιμάνια ενώ σε περιοχές μικρότερης πυκνότητας δημιουργούνται "μικροσυστάδες". Η υπόθεση μας είναι πως αυτό οφείλεται στον τρόπο με τον οποίο επιτρέπει ο OPTICS την μεταβλητή πυκνότητα στις συστάδες. Ο ελάχιστος αριθμός σημείων που πρέπει να έχει μια γειτονιά είναι σταθερός και ορίζεται από το $minPts$. Η μεταβλητή πυκνότητα προκύπτει από την δυναμική τιμή της παραμέτρου ϵ , που ορίζει το μέγεθος μιας γειτονιάς. Αυτό σημαίνει πως μπορούν να δημιουργηθούν συστάδες με πολύ διαφορετικά μεγέθη, κάτι που δεν είναι επιθυμητό για την συγκεκριμένη ανάλυση καθώς τα λιμάνια δεν διαφέρουν σε μέγεθος τόσο δραματικά μεταξύ τους.

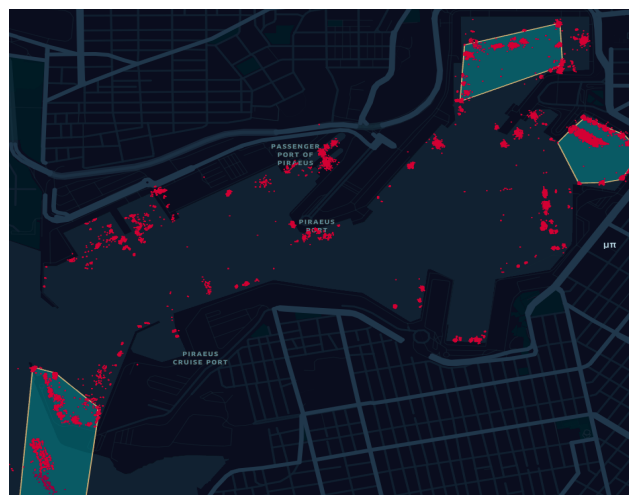
Φυσικά ο αλγόριθμος επιτρέπει τον περιορισμό του μεγέθους των συστάδων με την παράμετρο *maxEps* (μέγιστη τιμή του ϵ που θα ληφθεί υπόψιν). Είναι δυνατό να περιορίσουμε το ϵ ώστε να αποφύγουμε την δημιουργία μεγάλων συστάδων. Παρακάτω φαίνονται κάποια αποτελέσματα του OPTICS με περιορισμένη την τιμή ϵ . Τα πολύγωνα είναι χρωματισμένα βάσει των παραμέτρων του αλγορίθμου που τα δημιούργησαν:

OPTICS περιορισμένο *maxEps*, διαφορετικές τιμές *maxEps, minPts* βάσει χρώματος



Όλες οι εκτελέσεις του αλγορίθμου δυσκολεύονται να δημιουργήσουν μία μόνο συστάδα για ένα ολόκληρο λιμάνι και τείνουν να δημιουργούν μικρότερες συστάδες όταν το λιμάνι είναι μεγάλο. Αν παρατηρηθούν τα στίγματα που εισέρχονται στον αλγόριθμο μπορεί να φανεί ο λόγος:

Στίγματα για το λιμάνι του Πειραιά και μία εκτέλεση του OPTICS



Τα στίγματα των λιμανιών δεν είναι ίσα κατανεμημένα. Ένα λιμάνι αποτελείται από διάφορες περιοχές όπως terminals - περιοχές όπου δένουν τα καράβια για την φόρτωση και εκφόρτωση του εμπορεύματος. Αυτές οι περιοχές θα έχουν μεγαλύτερη πυκνότητα στιγμάτων. Βέβαια, το ε είναι αρκετά μεγάλο για να δημιουργηθεί μία ολόκληρη συστάδα για το λιμάνι. Ο λόγος που δεν γίνεται αυτό οφείλεται στην υλοποίηση του OPTICS στην sklearn. Όπως αναφέρθηκε, ο OPTICS δεν δημιουργεί από μόνος του συστάδες, αλλά βασίζεται σε έναν άλλο αλγόριθμο για να εξάγει τις συστάδες από το διάγραμμα γειτνίασης που δημιουργεί. Η βασική επιλογή εξαγωγής συστάδων της sklearn (xi) προσπαθεί να εξάγει τις καλύτερα δυνατές συστάδες, και δεδομένου ότι κάθε λιμάνι τείνει να αποτελείται από μικρές συστάδες με κενά μεταξύ τους, ο αλγόριθμος θεωρεί αυτήν την συσταδοποίηση ως την καλύτερη επιλογή.

Δύο πιθανές επιλογές για τον την δημιουργία συστάδων στο επίπεδο του λιμανιού λήφθηκαν υπόψη:

Αρχικά, μπορεί να χρησιμοποιηθεί διαφορετικός αλγόριθμος για την εξαγωγή συστάδων από το διάγραμμα γειτνίασης του OPTICS. Η sklearn επιτρέπει την χρήση του DBSCAN για την εξαγωγή συστάδων με την συνάρτηση `cluster_optics_dbscan`. Επειδή ο DBSCAN βρίσκει την μεγαλύτερη δυνατή συστάδα που ικανοποιεί την πυκνότητα που έχει δοθεί, η χρήση του μπορεί να επιτρέψει την δημιουργία μεγαλύτερων συστάδων. Παρ'όλα αυτά, δεν θεωρείται καλή λύση διότι όπως αναφέρεται και στο [documentation της συνάρτησης](#), οι διαφορές της `cluster_optics_dbscan` με τον `dbscan` είναι μικρές, και επειδή τα δεδομένα που χρησιμοποιούνται για τον `optics` είναι ένα δείγμα των αρχικών δεδομένων, η εκτέλεση του `dbscan` με όλα τα δεδομένα είναι καλύτερη.

Δεύτερη επιλογή είναι η χρήση της παραμέτρου `min_cluster_size` για να περιοριστεί η δημιουργία μικρότερων συστάδων. Η παράμετρος επιτρέπει τον ορισμό του ελάχιστου αριθμού σημείων που πρέπει να έχει μία συστάδα για να δημιουργηθεί. Δυστυχώς, αυτή επίσης δεν είναι μια καλή λύση καθώς αν το `min_cluster_size` δεν ικανοποιείται από τα terminals του Πειραιά, δεν θα ικανοποιείται από τα περισσότερα λιμάνια της Ελλάδας.

Ως αποτέλεσμα, καμία μεθοδολογία που δοκιμάστηκε δεν έδωσε ικανοποιητικά αποτελέσματα για τον OPTICS.

4.4 Εκτέλεση DBSCAN

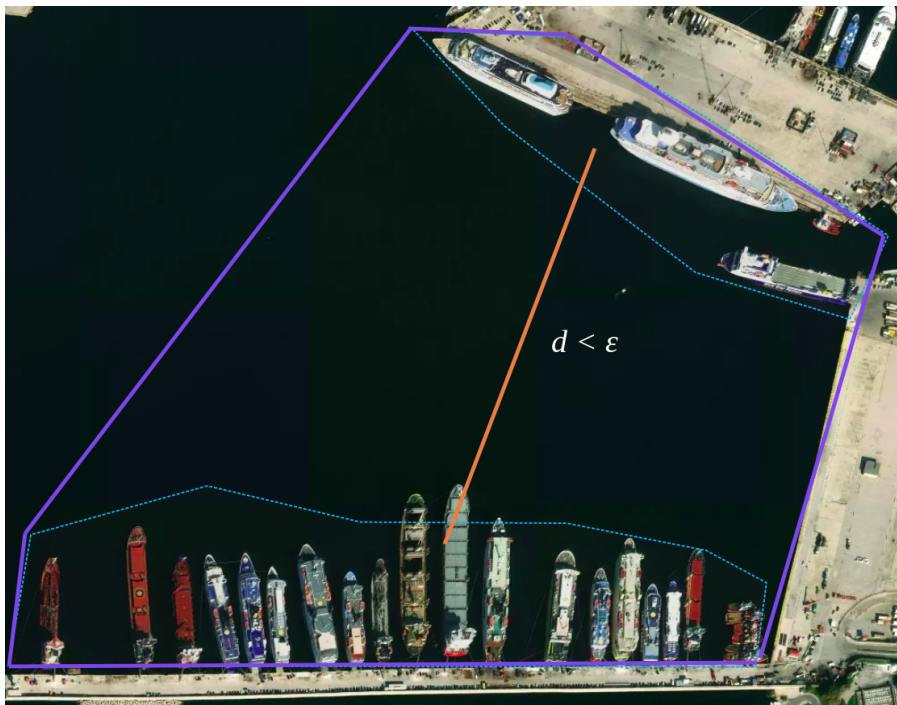
Ο δεύτερος αλγόριθμος που δοκιμάστηκε για τον εντοπισμό και την οριοθέτηση λιμανιών είναι ο DBSCAN. Ο DBSCAN είναι πιο ευαίσθητος στις παραμέτρους του συγκριτικά με τον OPTICS. Η επιλογή των παραμέτρων του, ε και `minPts`, είναι η σημαντικότερη απόφαση στην εκτέλεση του.

4.4.1 Επιλογή τιμής ε

Αρχικά, όπως αναφέρθηκε και στο κεφάλαιο 2.3.2, επειδή η ανάλυση γίνεται με γεωγραφικά στίγματα, για την τιμή του ε πρέπει να χρησιμοποιηθεί η γεωδαισιακή απόσταση και όχι η ευκλείδεια. Για τον υπολογισμό της γεωδαισιακής απόστασης μπορεί να χρησιμοποιηθεί η εξίσωση haversine, η οποία υποστηρίζεται από την υλοποίηση του DBSCAN της sklearn. Επειδή η εξίσωση haversine χρειάζεται τις τιμές σε ακτίνια, πρέπει να γίνει η κατάλληλη μετατροπή όλων των παραμέτρων. Για την τιμή ε , ο υπολογισμός της απόστασης σε ακτίνια μπορεί να γίνει αν διαιρεθεί η απόσταση ε (σε χιλιόμετρα) με την ακτίνα της γης, που είναι περίπου 6371 χιλιόμετρα. Για το υπόλοιπο κεφάλαιο όπου γίνεται αναφορά κάποιας απόστασης ε σε χιλιόμετρα η παράμετρος που δέχεται ο DBSCAN είναι $\varepsilon/6371$.

Για την επιλογή της κατάλληλης τιμής της παραμέτρου ε , πρέπει να περιοριστούν οι πιθανές τιμές της. Υπενθυμίζεται πως το ε ορίζει την ακτίνα μιας γειτονιάς. Μια γειτονιά μπορεί να οριστεί ως μία μικρή συστάδα με μέγιστη ακτίνα ε και κέντρο το κεντρικό σημείο της. Όταν δύο γειτονιές βρίσκονται σε απόσταση $< \varepsilon$ μεταξύ τους, ενώνονται και δημιουργούν μία μεγαλύτερη συστάδα. Στα λιμάνια, τα κεντρικά σημεία-γειτονιές θα δημιουργηθούν στα σημεία μεγαλύτερης πυκνότητας, τα berth και terminal των λιμανιών. Στόχος μας είναι οι μικρότερες συστάδες να ενωθούν και να δημιουργήσουν μία μεγαλύτερη συστάδα για ολόκληρο το λιμάνι. Αυτό σημαίνει πως η απόσταση μεταξύ δύο οποιονδήποτε terminal πρέπει να είναι μικρότερη του ε .

Δύο μικρότερες συστάδες δημιουργούν μία καινούργια όταν η απόσταση μεταξύ τους είναι $< \varepsilon$



Αντίστοιχα, το ε πρέπει να είναι μικρότερο της απόστασης μεταξύ δύο λιμανιών ώστε να μην δημιουργηθεί συστάδα που να περιέχει δύο ή περισσότερα λιμάνια, όταν αυτά βρίσκονται κοντά μεταξύ τους. Αυτό περιορίζει τις επιθυμητές τιμές της παραμέτρου ε ως εξής:

Απόσταση μεταξύ terminals $< \varepsilon < \text{απόσταση μεταξύ λιμανιών}$

Αυτό ο περιορισμός είναι αρκετός ώστε να επιτρέψει την χρήση brute force για την βέλτιστη επιλογή της τιμής της παραμέτρου ε .

4.4.2 Επιλογή τιμής $\minPts$

Το $\minPts$ είναι πιο δύσκολο να περιοριστεί σε συγκεκριμένες τιμές. Διαισθητικά, η παράμετρος $\minPts$ ορίζει το πόσο συντηρητική είναι η ανάλυση συστάδων. Μικρότερες τιμές στο $\minPts$ επιτρέπουν τον εντοπισμό περισσότερων συστάδων αλλά έχουν ως αποτέλεσμα την δημιουργία συστάδων σε σημεία που δεν υπάρχουν λιμάνια. Αντίστοιχα, μεγαλύτερες τιμές στο $\minPts$ σημαίνει πως περισσότερα σημεία θα θεωρηθούν ως θόρυβος και ως αποτέλεσμα, υπάρχει μεγαλύτερη πιθανότητα να μην εντοπιστούν μικρότερα λιμάνια.

Η επιλογή της τιμής *minPts* στον DBSCAN συνήθως γίνεται βάσει της γνώσης του αναλυτή πάνω στο πεδίο στο οποίο γίνεται η ανάλυση, αλλά στο συγκεκριμένο πρόβλημα είναι δύσκολο να γίνει μια επιλογή που να θεωρείται αντιπροσωπευτική του αριθμού πλοίων που εξυπηρετεί ένα λιμάνι σε κάποιο χρονικό διάστημα. Η μόνη επιλογή είναι να γίνει δοκιμή διάφορων πιθανών τιμών και σύγκριση των αποτελεσμάτων τους.

4.4.3 Εκτέλεση Αλγορίθμου

Ένας τρόπος για την επιλογή των παραμέτρων του DBSCAN είναι να γίνει προσπάθεια εντοπισμού κάποιου σημείου καμπής, όπου βάσει κάποιων παραμέτρων ο αλγόριθμος εντοπίζει τα περισσότερα λιμάνια και δημιουργεί όσο το δυνατόν λιγότερες συστάδες σε λάθος σημεία.

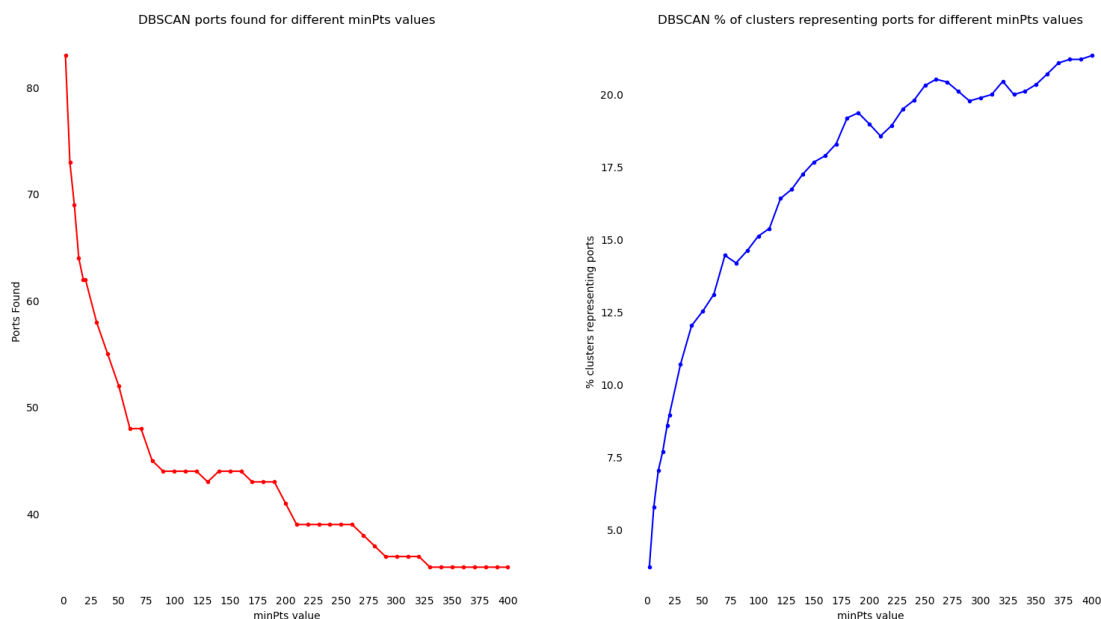
Το σημείο καμπής μπορεί να εντοπιστεί με την χρήση δύο μεταβλητών για την αξιολόγηση των αποτελεσμάτων της κάθε εκτέλεσης του αλγορίθμου:

1. Αριθμός λιμανιών που εντοπίστηκαν.
2. Ποσοστό συστάδων που αντιστοιχούν σε κάποιο λιμάνι.

Για την εξαγωγή των παραπάνω τιμών χρησιμοποιήθηκε η λίστα των λιμανιών της Ελλάδας του World Port Source και ένα script της python (appendix: Αξιολόγηση Αποτελεσμάτων DBSCAN).

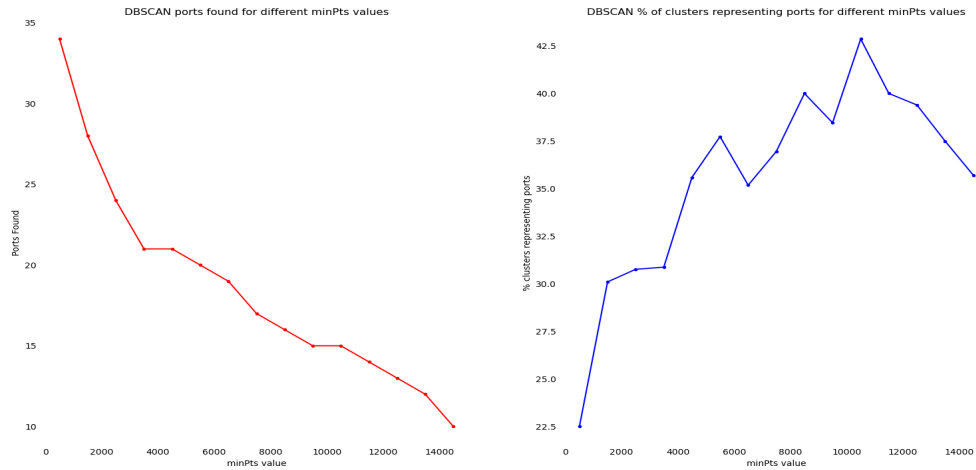
Τα αποτελέσματα φαίνονται στο παρακάτω διάγραμμα:

Αξιολόγηση αποτελεσμάτων του DBSCAN - μικρές τιμές *minPts*, $\epsilon=0.55\text{km}$



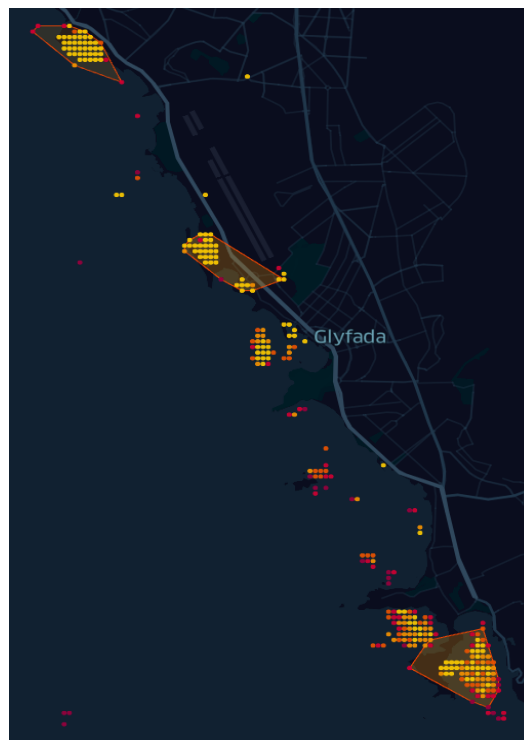
Ένα πρόβλημα που προκύπτει από το διάγραμμα είναι πως ακόμα και μεγάλες τιμές *minPts* εντοπίζουν πολλές συστάδες που δεν αντιστοιχούν σε λιμάνια (~80%). Εν μέρει το πρόβλημα μπορεί να οφείλεται στα αγκυροβόλια καθώς είναι περιοχές μεγάλης πυκνότητας καραβιών με χαμηλή ταχύτητα και δεν θεωρούνται κομμάτι του λιμανιού από τον αλγόριθμο. Αυτό όμως, δεν είναι αρκετό για να δικαιολογήσει το μεγάλο ποσοστό αυθαίρετων συστάδων. Το μοτίβο συνεχίζεται ακόμα και σε μεγαλύτερη κλίμακα όπου ακόμα και όταν ο αλγόριθμος εντοπίζει δέκα μόνο συστάδες, οι μισές από αυτές δεν ανήκουν σε κάποιο λιμάνι:

Αξιολόγηση αποτελεσμάτων του DBSCAN - μεγάλες τιμές minPts, $\epsilon=0.55\text{km}$



Από τον χάρτη φαίνεται πως τουλάχιστον κάποιες από τις αυθαίρετες συστάδες αποτελούν λιμάνια τα οποία δεν υπάρχουν στην λίστα του world port source, δεδομένου ότι δεν μπορεί να υπάρξει σημείο με χιλιάδες σταματημένα καράβια χωρίς λόγο. Τα αποτελέσματα του αλγορίθμου, τουλάχιστον στο κομμάτι του εντοπισμού, είναι ικανοποιητικά.

Αυθαίρετες συστάδες με minPts=10000



Δυστυχώς η αξιολόγηση του πραγματικού αριθμού των λιμανιών που εντοπίζονται είναι δύσκολη χωρίς μία πλήρη πηγή που να περιέχει όλα τα λιμάνια της Ελλάδας. Σε αυτό το σημείο, θα μπορούσε να γίνει η υλοποίηση ενός προγράμματος για τον εντοπισμό των σημείων που μεγιστοποιούνται οι μεταβλητές βάσει των οποίων γίνεται η αξιολόγηση. Δεδομένου όμως ότι το world port source δεν είναι μία πλήρης πηγή, αυτή η μέθοδος δεν εγγυάται τα καλύτερα αποτελέσματα και θα οδηγούσε σε overfit του αλγορίθμου στο world port source.

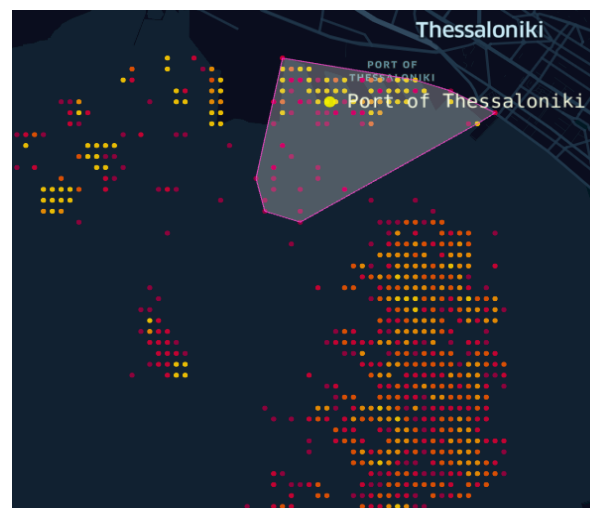
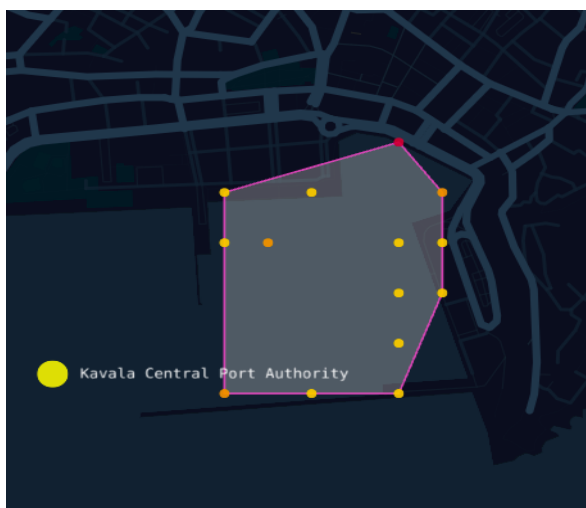
Στόχος μας ήταν η δημιουργία μιας μεθοδολογίας που να μπορεί να δώσει ικανοποιητικά αποτελέσματα με την διαθέσιμη πηγή χωρίς να βασίζεται σε αυτήν, έτσι ώστε μια μελλοντική πληρέστερη πηγή να μπορεί να δώσει καλύτερα αποτελέσματα. Βάσει αυτού και όλων των πληροφοριών που παρουσιάστηκαν σε αυτό το κεφάλαιο, ακολουθεί η τελική μεθοδολογία που χρησιμοποιήθηκε για τον εντοπισμό και την οριοθέτηση των λιμένων:

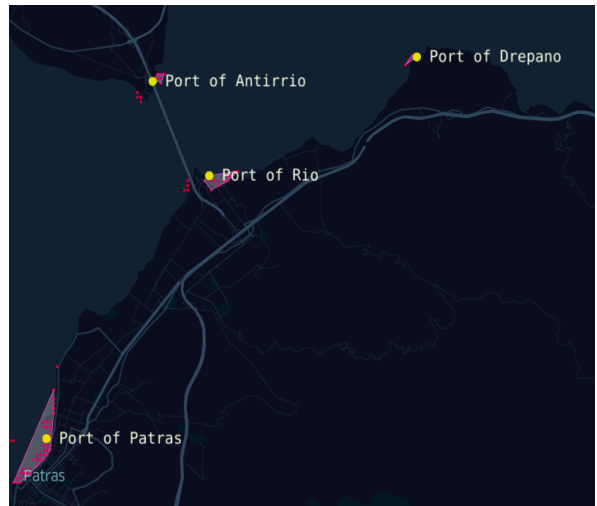
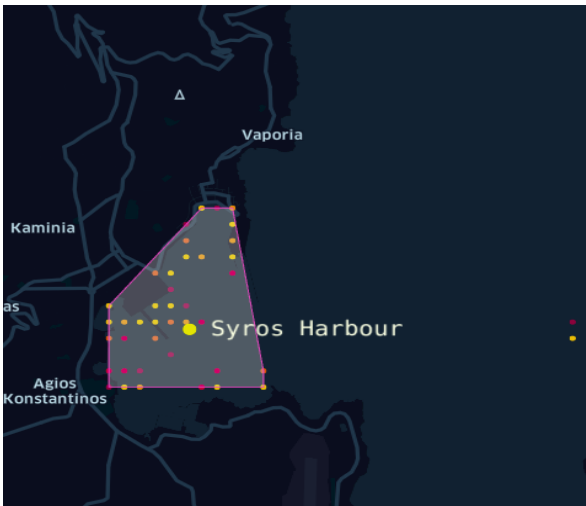
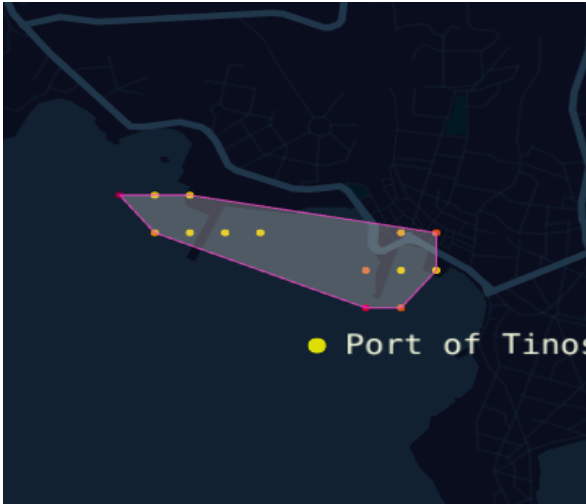
1) Χρήση χαμηλής τιμής *minPts*. Όπως φαίνεται και στα διαγράμματα, μόνο οι χαμηλότερες τιμές στο *minPts* επιτρέπουν στον αλγόριθμο να εντοπίσει τα περισσότερα λιμάνια.

2) Φιλτράρισμα των πολυγώνων βάσει κάποιας λίστας λιμανιών. Για κάθε πολύγωνο, γίνεται ο εντοπισμός του κοντινότερου λιμανιού και η ανάθεση του σε αυτό. Τα πολύγωνα που δεν έχουν κοντά λιμάνια αφαιρούνται από τα αποτελέσματα. Μια πληρέστερη λίστα θα έχει ως αποτέλεσμα τον εντοπισμό περισσότερων λιμανιών χωρίς να γίνει αλλαγή στον αλγόριθμο.

Αυτή η μεθοδολογία επιτρέπει την οριοθέτηση όλων των μεγάλων λιμανιών δεδομένης της τοποθεσίας του ενώ επιτρέπει την οριοθέτηση και μικρότερων λιμανιών. Επειδή ο αλγόριθμος δημιουργεί τις συστάδες χωρίς επίβλεψη πριν γίνει το φιλτράρισμα τους, έχει το πλεονέκτημα ότι δημιουργεί συστάδες μόνο σε λιμάνια στα οποία υπάρχει κινητικότητα και υπάρχει λόγος παρακολούθησής τους.

Στις παρακάτω εικόνες φαίνεται ένα δείγμα των αποτελεσμάτων του αλγορίθμου με *minPts=10* και *Eps=0.55km*. Τα αποτελέσματα είναι καλύτερα για λιμάνια που έχουν μεγαλύτερη κινητικότητα καθώς η οριοθέτηση είναι δύσκολη για λιμάνια τα οποία δεν έχουν αρκετά στίγματα πλοίων στην περιοχή τους.





(κενή σελίδα)

5. Παρακολούθηση Λιμένων

Τα αποτελέσματα της ανάλυσης συστάδων μπορούν να χρησιμοποιηθούν για την παρακολούθηση των λιμανιών που εντοπίστηκαν. Όπως αναφέρθηκε, για το geofencing χρησιμοποιήθηκε το tile38.

Για κάθε καινούργιο στίγμα που εισάγεται στην βάση του, το tile38 εντοπίζει την θέση του σε σχέση με τα πολύγωνα που παρακολουθούνται και επιστρέφει το ανάλογο γεγονός εντοπισμού. Υπάρχουν διάφορα γεγονότα εντοπισμού που υποστηρίζονται από το tile38:

- **inside:** Όταν ένα αντικείμενο εντοπίστηκε μέσα στην περιοχή.
- **outside:** Όταν ένα αντικείμενο εντοπίστηκε έξω από την περιοχή.
- **enter:** Όταν ένα αντικείμενο που πριν βρισκόταν έξω από την περιοχή, μπήκε μέσα σε αυτήν.
- **exit:** Όταν ένα αντικείμενο που πριν βρισκόταν μέσα στην περιοχή, βγήκε έξω από αυτήν.
- **cross:** Όταν ένα αντικείμενο που πριν βρισκόταν έξω από το περιοχή, μπήκε και βγήκε από αυτήν.

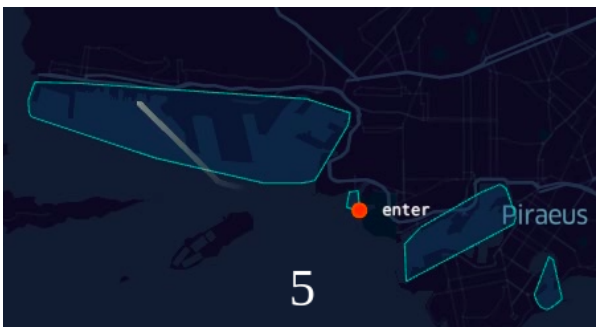
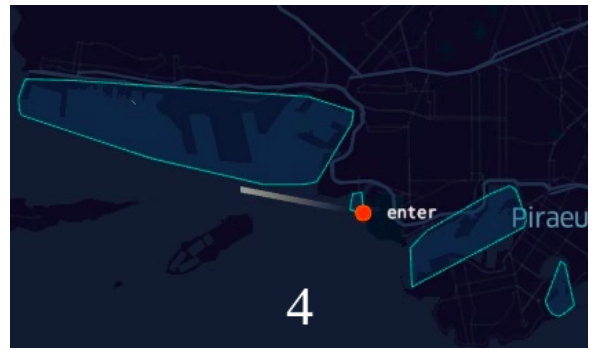
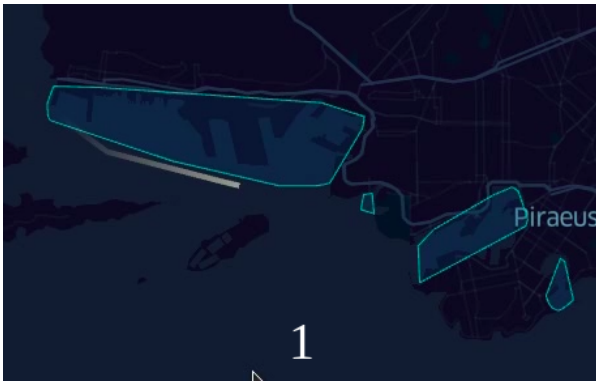
Για τους στόχους της εργασίας, τα enter και exit events είναι τα πιο σημαντικά. Τα υπόλοιπα events προσθέτουν μεγάλο όγκο στα δεδομένα χωρίς να παρέχουν πολύ πληροφορία. Για την υλοποίηση του προγράμματος παρακολούθησης χρησιμοποιήθηκε ο κώδικας geofencing (βλ. Appendix).

Ο κώδικας διαβάζει την λίστα με τα πολύγωνα και δημιουργεί ένα κανάλι για κάθε πολύγωνο για τα enter και exit events. Έπειτα γίνεται εγγραφή σε κάθε κανάλι, οπότε το πρόγραμμα είναι και ο publisher και ο subscriber στην συγκεκριμένη περίπτωση.

Για την δοκιμή του προγράμματος έγινε η υλοποίηση ενός μικρού script στην python που διαβάζει τα στίγματα AIS από τα αρχικά δεδομένα και τα προσθέτει στην βάση για να δημιουργηθούν τα γεγονότα παρακολούθησης. Για την επίμονη αποθήκευση των γεγονότων εντοπισμού, έγινε εγγραφή του γεγονότος μαζί με κάποια στοιχεία για αυτό σε ένα εξωτερικό αρχείο μορφής json.

Επειδή δεν υπάρχει πηγή σύγκρισης, τα αποτελέσματα της παρακολούθησης είναι δύσκολο να αξιολογηθούν αντικειμενικά. Γίνεται όμως, με την χρήση του kepler, να οπτικοποιηθούν.

Το kepler μπορεί να δεχτεί ένα σύνολο από στίγματα για ένα καράβι μαζί με την χρονική στιγμή που εντοπίστηκαν και να δημιουργήσει μία κινούμενη γραμμή που αναπαριστά την πορεία του καραβιού. Επίσης μπορεί να δεχτεί δεδομένα για events που γίνονται κάποια χρονική στιγμή σε κάποια τοποθεσία. Δημιουργήθηκαν δύο καινούργια σετ δεδομένων, ένα με το μονοπάτι ενός καραβιού και ένα με τα enter events του. Τα αποτελέσματα φαίνονται στις παρακάτω εικόνες:



6. Αξιολόγηση Αποτελεσμάτων

Σαν αλγόριθμος μηχανικής μάθησης, πρέπει να γίνει κατάλληλη αντικειμενική αξιολόγηση των αποτελεσμάτων του. Κοινά κριτήρια για την αξιολόγηση των αλγορίθμων επιτρέπουν την άμεση σύγκριση της αποτελεσματικότητας των μεθοδολογιών με στόχο την εύρεση του βέλτιστου αλγορίθμου που δίνει τα καλύτερα αποτελέσματα για το πρόβλημα.

Δεδομένου ότι ο αλγόριθμος προσπαθεί να εντοπίσει και να οριοθετήσει το σύνολο των λιμανιών στην Ελλάδα, πηγές που καταγράφουν λιμάνια όπως αυτήν που χρησιμοποιήθηκε κατά την διάρκεια της ανάλυσης μπορούν να χρησιμοποιηθούν για την αξιολόγηση των συστάδων που δημιούργησε ο αλγόριθμος. Εδώ, το ποσοστό των λιμανιών που εντοπίστηκαν είναι πιο σημαντικό από τον αριθμό των αυθαίρετων συστάδων. Όπως φάνηκε και στην ανάλυση, πολλές από τις αυθαίρετες συστάδες που εντοπίστηκαν αποτελούν λιμάνια τα οποία δεν περιλαμβάνονται στην πηγή. Η καταγραφή τους ως λάθος συστάδες θα διαστρεβλώσει την αποτελεσματικότητα του αλγορίθμου.

Αυτό δεν σημαίνει πως 103 είναι ο ιδανικός αριθμός συστάδων για την ανάλυση συστάδων. Όπως αναφέρθηκε, ένα σημαντικό μέρος των μικρότερων λιμανιών της Ελλάδας δεν περιέχουν καθόλου στίγματα AIS κοντά τους και ως αποτέλεσμα ο εντοπισμός του είναι αδύνατος με την μεθοδολογία της εργασίας. Επειδή δεν είναι γνωστό ποιο μέρος αυτού του περιορισμού είναι λόγω λάθους των δεδομένων και ποιο είναι αντιπροσωπευτικό της πραγματικότητας, ο ακριβής ορισμός του ιδανικού αριθμού συστάδων είναι δύσκολος.

Με την χρήση του World Port Source προκύπτουν τα εξής αποτελέσματα:

Για τα 103 ελληνικά λιμάνια που καταγράφονται στην πηγή, ο αλγόριθμος συνολικά εντόπισε και οριοθέτησε 60 από αυτά. Για τα 8 λιμάνια που καταγράφονται στην πηγή ως "μεσαιά" η μεγαλύτερα, ο αλγόριθμος εντόπισε και τα 8.

Η χρήση του πεδίου της κατάστασης πλοήγησης δεν δίνει καλύτερα αποτελέσματα. Ο αλγόριθμος εντοπίζει 8/8 από τα μεγαλύτερα λιμάνια της Ελλάδας και μόλις 39 λιμάνια γενικότερα. Τα αποτελέσματα δείχνουν ότι ίσως η κατάσταση πλοήγησης δεν συμπληρώνεται σωστά από το πλήρωμα κάθε φορά που το καράβι αγκυροβολεί σε ένα λιμάνι.

Για τα εμπορικά πλοία ο αλγόριθμος εντοπίζει 49 λιμάνια που εξυπηρετούν εμπορικά πλοία, αλλά λόγω του μειωμένου όγκου των δεδομένων τα πολύγωνα είναι λιγότερο ολοκληρωμένα στα μικρότερα λιμάνια, και τα αποτελέσματα είναι λιγότερο χρήσιμα. Το ίδιο ισχύει και για τον συνδυασμό εμπορικών πλοίων που αγκυροβολούν με την χρήση της κατάστασης πλοήγησης.

Μια διαφορετική πληροφορία που έχει αξία, είναι το ποσοστό των ελληνικών λιμανιών που μπορεί να εντοπίσει ο αλγόριθμος από τα λιμάνια τα οποία έχουν στίγματα AIS κοντά τους. Από τα 103 ελληνικά λιμάνια, τα 13 από αυτά δεν έχουν στίγματα κοντά τους. Από τα 90 ελληνικά λιμάνια που έχουν εξυπηρετήσει κάποιο καράβι με AIS, ο αλγόριθμος εντοπίζει τα 60. Τα υπόλοιπα λιμάνια που δεν εντοπίστηκαν ανήκουν σε μία από δύο κατηγορίες:

- Έχουν <10 στίγματα κοντά τους και η περιοχή δεν θεωρείται αρκετά πυκνή από τον αλγόριθμο για την δημιουργία συστάδας.
- Έχουν αρκετά στίγματα κοντά τους αλλά σε ένα μόνο γεωγραφικό σημείο, οπότε η δημιουργία πολυγώνου είναι αδύνατη

7. Συμπεράσματα

7.1 Πιθανές Επεκτάσεις

Η εργασία αποτελεί μία εισαγωγική ανάλυση πάνω στον τομέα αξιοποίησης των δεδομένων του αυτόματου συστήματος αναγνωρίσεως και υπάρχουν αρκετά περιθώρια βελτίωσης και επέκτασης. Αρχικά, ένα από τα προβλήματα που παρατηρήθηκε ήταν η μεγάλη διαφορά στον αριθμό των πλοίων που εξυπηρετούνται από λιμάνι σε λιμάνι. Ακόμα και με τον OPTICS, δεν καταφέραμε να εντοπίσουμε με επιτυχία τα περισσότερα λιμάνια χωρίς την δημιουργία αυθαίρετων συστάδων. Πιθανώς, ένας αλγόριθμος ανάλυσης συστάδων για τον εντοπισμό συστάδων μεταβλητής πυκνότητας και σταθερού γεωγραφικού μεγέθους να μπορεί να λύσει αυτό το πρόβλημα.

Ένα άλλο κομμάτι που θα μπορούσε να βελτιωθεί είναι ο εντοπισμός και η οριοθέτηση. Η εργασία περιορίζεται στον εντοπισμό και στην οριοθέτηση μόνο των λιμανιών. Μία πιο λεπτομερής ανάλυση θα μπορούσε να εντοπίσει και τις μικρότερες συστάδες που βρίσκονται στα λιμάνια και να οριοθετήσει τα berths του λιμανιού. Αυτή η οριοθέτηση παρέχει πιο λεπτομερή πληροφορία που μπορεί να χρησιμοποιηθεί για την πραγματικού χρόνου παρακολούθηση των ελεύθερων θέσεων σε ένα λιμάνι, κάτι που μπορεί να βελτιώσει την αποδοτικότητα των πλοίων και των λιμανιών.

Μία άλλη πιθανή επέκταση αφορά το πρόγραμμα παρακολούθησης λιμένων. Για την εξαγωγή οικονομικά σημαντικής πληροφορίας, η δημιουργία γεγονότων εντοπισμού κάθε φορά που εντοπίζεται ένα καράβι σε ένα λιμάνι δεν είναι αρκετή. Όπως φάνηκε και στην εργασία των Cerdeiro *et al.*(2020), η εξαγωγή των port calls είναι μια πολύπλοκη διαδικασία που απαιτεί την δικιά της ανάλυση. Μία επέκταση της εργασίας θα μπορούσε να χρησιμοποιήσει τα αποτελέσματα της για τον εντοπισμό των port calls και την αξιολόγηση τους βάσει κάποιας επίσημης πηγής.

Τέλος, η εργασία περιορίστηκε στην χρήση δύο μόνο αλγορίθμων για την ανάλυση συστάδων. Η χρήση διαφορετικών αλγορίθμων όπως ο HDBSCAN και η σύγκριση τους με τα αποτελέσματα αυτής της εργασίας μπορεί να βοηθήσει στην δημιουργία μιας καλύτερης μεθοδολογίας.

7.2 Γενικά Συμπεράσματα

Η εργασία δείχνει, όπως έχουν δείξει και άλλες εργασίες στο παρελθόν ότι τα δεδομένα AIS μπορούν, με την χρήση αλγορίθμων μηχανικής μάθησης, να χρησιμοποιηθούν για τον εντοπισμό και την οριοθέτηση λιμένων. Τουλάχιστον για τους αλγόριθμους που δοκιμάστηκαν, ο DBSCAN δίνει ακόμα τα καλύτερα αποτελέσματα, αλλά έχει προβλήματα που περιορίζουν την αποτελεσματικότητά του. Αν και τα αποτελέσματα των παρόμοιων εργασιών στο θέμα δεν είναι διαθέσιμα δωρεάν, είναι αμφίβολο πως τα αποτελέσματα της εργασίας μας είναι τα καλύτερα. Στόχος της εργασίας δεν ήταν να λύσει το πρόβλημα της βέλτιστης οριοθέτησης των λιμανιών, ένα πρόβλημα το οποίο παραμένει άλυτο. Στόχος της εργασίας ήταν να μπορεί να χρησιμοποιηθεί ως βάση για περαιτέρω ανάλυση πάνω στο θέμα. Στο πνεύμα της γενικότερης βελτίωσης των ορίων των λιμανιών στα πληροφοριακά συστήματα, κάνουμε τον βασικό κώδικα και τα αποτελέσματα μας δημόσια διαθέσιμα.

Βιβλιογραφία

Yang, D., Wu, L., Wang, S., Jia, H., & Li, K. X. (2019). How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications. *Transport Reviews*, 1–19.

<https://doi.org/10.1080/01441647.2019.1649315>

Harati-Mokhtari, A., Wall, A., Brooks, P., & Wang, J. (2007). Automatic Identification System (AIS): Data Reliability and Human Error Implications. *Journal of Navigation*, 60(03), 373.

<https://doi.org/10.1017/s0373463307004298>

Restrepo, M. (2020) 'Clustering Ship Data to Identify Port Boundaries', General Atomics CCRI, 2nd December.

Available at: <https://ccri.com/clustering-ship-data-to-identify-port-boundaries/> (Accessed: 27 September 2021).

National Geospatial-Intelligence Agency (2019). World Port Index 2019. Available at:

<https://msi.nga.mil/api/publications/download?key=16694622/SFH00000/Pub150bk.pdf&type=view> (Accessed: 27 September 2021)

Cerdeiro, D.A., Komaromi A., Liu Y., Saeed M. (2020) World Seaborne Trade in Real Time: A Proof of Concept for Building AIS-based Nowcasts from Scratch. Available

at: <https://www.imf.org/en/Publications/WP/Issues/2020/05/14/World-Seaborne-Trade-in-Real-Time-A-Proof-of-Concept-for-Building-AIS-based-Nowcasts-from-49393> (Accessed 28 September)

Makles, A. (2012) 'Stata tip 110: How to get the optimal k-means cluster solution', *The Stata Journal*, Number 2, pp.347–351

Ester, M., H.-P. Kriegel, J. Sander, X. Xu. (1996). “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*.

Ankerst, M. et al. (1999) “OPTICS: Ordering points to identify the clustering structure,” *SIGMOD record*, 28(2), pp. 49–60.

Moreira, A., Santos, M. Y. (2007) Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points, Uminho.pt. Available at:

https://repositorium.sdum.uminho.pt/bitstream/1822/6429/1/ConcaveHull_ACM_MYS.pdf (Accessed: September 28, 2021).

Liu, T., Gao, S., Chu, X., & Lu, C. (2017). Storing and querying AIS data in HBase. 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS). <https://doi.org/10.1109/acirs.2017.7986071>

IBM (2020) What is Machine Learning? . Available at: <https://www.ibm.com/cloud/learn/machine-learning> (Accessed: 29 September 2021).

Erman, J., Arlitt, M., & Mahanti, A. (2006). Traffic classification using clustering algorithms. *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data - MineNet '06*. <https://doi.org/10.1145/1162678.1162679>

Appendinx

Αξιολόγηση Αποτελεσμάτων DBSCAN

```
import pandas as pd
import geopandas as gpd
from shapely import wkt
from shapely.geometry import Point, Polygon
from sklearn.cluster import DBSCAN
import numpy as np
import matplotlib.pyplot as plt

min_samples_values = list(range(2,20,4)) + list(range(20,401,10)) #test for values up to 400
eps = 0.55
df = pd.read_csv('weighted_data.csv')
ports = pd.read_csv('ports_small.csv')

def convex_hull(gdf):
    return gdf.unary_union.convex_hull

#given a single cluster, find the closest port to it and assign it to the cluster
#given that points are rounded to 3 decimals, there will be multiple points in a single location
#a cluster will be a single point if all of it's points are in one location
#a cluster will be a linestring if all of it's points are in two locations
def find_closest_port(cluster):
    cluster['distance']=1000
    distance = cluster['distance']

    for index,port in ports.iterrows():
        point = Point(port['longitude'],port['latitude'])
        if cluster.geometry.distance(point)==0:
            cluster['distance']=0
            cluster['port']=port['port']
            return cluster

    if cluster.geometry.type=='Polygon':
        distance = cluster.geometry.boundary.distance(point)
    else:
        distance = cluster.geometry.distance(point)
    if distance < cluster['distance']:
        cluster['distance']=distance
        cluster['port']=port['port']
    return cluster
```

```

l = []
for min_samples in min_samples_values:
    #haversine formula is used to calculate geodesic distance
    #haversine gives shortest distance between two points on earth
    #it requires it's values to be in radians
    #eps represents distance in km. Divide by 6371 (earth radius) to get the distance in radians
    db =
    DBSCAN(eps=eps/6371.,min_samples=min_samples,algorithm='ball_tree',metric='haversine').fit(np.radians(df[['
latitude','longitude']]),sample_weight=df['weight'])
    df['label'] = db.labels_

    gdf = gpd.GeoDataFrame(df,geometry=gpd.points_from_xy(df.longitude,df.latitude))
    gdf = gdf.groupby('label')['geometry'].agg(convex_hull)
    gdf = gdf[1:].reset_index() #the first row represents the polygon of the unclustered points, remove it

    gdf = gdf.apply(lambda x: find_closest_port(x),axis=1)
    found = gdf[gdf['distance']<0.03]
    found = found.sort_values(by='distance').drop_duplicates(subset='port')
    l.append([min_samples,len(gdf.index),len(found.index)])

l = list(zip(*l))

plt.figure(figsize=(8,3))
plt.subplot(1,2,1)
plt.box(False)
plt.plot(l[0],l[2],color='red',marker='.')
plt.xticks(range(0,401,25))
plt.xlabel('minPts value')
plt.ylabel('Ports Found')
plt.title('DBSCAN ports found for different minPts values')
plt.tick_params(axis='both', length = 0)

plt.subplot(1,2,2)
plt.box(False)
plt.xticks(range(0,401,25))
plt.tick_params(axis='both', length = 0)
plt.ylabel('% clusters representing ports')
plt.xlabel('minPts value')
plt.title('DBSCAN % of clusters representing ports for different minPts values')
plt.plot(l[0],[x/y*100 for x,y in zip(l[2],l[1])],color='blue',marker='.')
plt.show()

```

Geofencing

```
import pandas as pd
import geopandas as gpd
import redis
import shapely.geometry
from shapely import wkt
import geojson
import shapely as shp
import time
import json

client = redis.Redis(host='127.0.0.1', port=9851) #connect to tile38 server
p = client.pubsub()

with open('ports.geojson') as f:
    gj = geojson.load(f)

#for each port, create a channel listening to enter and exit events for it and then subscribe to it
for port in gj['features']:
    n = port['properties']['port']
    polygon=port['geometry']

client.execute_command("SETCHAN",n,"WITHIN","ship","FENCE","DETECT","enter,exit","OBJECT",str(polygon))
p.subscribe(name)

#infinite loop that constantly listens for new events. Whenever a new event is detected, write it to an external file.
while True:
    message = p.get_message()
    if message:
        if message['type']=='subscribe':
            print('Subscribed to '+message['channel'].decode('utf-8'))
            continue

        data = json.loads(message['data'].decode('utf-8'))
        port = message['channel'].decode('utf-8')
        print(data['detect'] + ' event for ship '+data['id']+ ' at ' + port + '(' + str(data['fields']['timestamp'])+')')
        f = open('results1.txt','a')
        f.write(str(message)+"\n")
        f.close()
```

Keppler Animation

```
import pandas as pd
import geojson
from dateutil import parser
import geojson
import json
from datetime import datetime

def convert_to_unix(timestamp):
    timestamp = datetime.strptime(timestamp, '%Y-%m-%d %H:%M:%S%z')
    return timestamp.timestamp()

def to_kepler(df):
    print(df)
    mmsi = int(df['mmsi'].iloc[0])

    coordinates = [] #list of lists representing a LineString. Each list
    contains a single point at a certain time.

    for index, row in df.iterrows():
        coordinates.append([row['longitude'], row['latitude'], 0, row['timestamp']])

    dic = {}
    dic['type'] = 'FeatureCollection'
    dic['features'] = [{
        'type': 'Feature',
        'properties': {'mmsi': mmsi},
        'geometry': {
            'type': 'LineString',
            'coordinates': coordinates
        }
    }]
    return dic

df = pd.read_csv('trip_data.csv')
unique_mmsi = set(df['mmsi'])
df = df.sort_values(by='timestamp')
df['timestamp'] = df.apply(lambda x: convert_to_unix(x['timestamp']), axis=1)
print(df)

maindf = pd.DataFrame(columns=['mmsi', 'geometry'])
```

```
i=0
for mmsi in unique_mmsi:
    single_ship_data = df[df.mmsi==mmsi]
    dic = to_keppler(single_ship_data)
    dic = json.dumps(dic)
    maindf.loc[i]= [mmsi]+[dic]
    i=i+1

maindf.to_csv('data.csv',index=False)
```

Αποτελέσματα Πολυγώνων

https://github.com/aMimikyu/AIS-geofencing/blob/main/ports_with_polygons.csv