ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Ψηφιακών Συστημάτων

Π.Μ.Σ. «Πληροφοριακά Συστήματα και Υπηρεσίες»

Ειδίκευση: «Μεγάλα Δεδομένα και Αναλυτική»

Διπλωματική Εργασία

Εντοπισμός δρομολογίου από τροχιά πλοίου σε πραγματικό χρόνο

Θεόφιλος Κωνσταντίνος Νεκτάριος | ΜΕ2012

Επιβλέπων: Χρήστος Δουλκερίδης

Πειραιάς, Φεβρουάριος 2023

Περίληψη

Στην παρούσα διπλωματική εργασία, αναπτύξαμε μια εφαρμογή η οποία λαμβάνει και επεξεργάζεται σε πραγματικό χρόνο μηνύματα που μεταξύ άλλων περιέχουν τις γεωγραφικές θέσεις κινούμενων πλοίων. Για κάθε πλοίο, έχοντας υπόψη ένα σύνολο προκαθορισμένων δρομολογίων, προσπαθεί να αντιστοιχίσει την τροχιά του με το δρομολόγιο εκείνο το οποίο είναι πιο πιθανό να ακολουθείται εκείνο το χρονικό διάστημα από το πλοίο. Για να γίνει αυτό, γίνεται χρήση μιας μετρικής απόστασης και βαθμολόγησης των εναλλακτικών δρομολογίων. Η εφαρμογή υλοποιήθηκε στην πλατφόρμα παράλληλης επεξεργασίας μεγάλων δεδομένων Apache Spark, και συγκεκριμένα στο εργαλείο Apache Spark Streaming.

Λέξεις-κλειδιά: Automatic Identification System (A.I.S.), πλοίο, γεωγραφική θέση, τροχιά, δρομολόγιο, ευκλείδεια απόσταση, παράλληλη επεξεργασία, ροή δεδομένων

Abstract

In this thesis, we developed a software application which receives and processes, in real time, messages from moving vessels. The messages contain, among other parameters, the geographical position of the vessel. For each vessel, the application considers a set of predetermined routes and tries to match its trajectory with the route that it is most likely that the vessel is currently following. To do that, a distance measure and scoring system is used. The application was developed using the general-purpose parallel processing Big Data framework Apache Spark, and specifically its component called Apache Spark Streaming.

Keywords: Automatic Identification System (A.I.S.), vessel, geographical position, trajectory, route, Euclidean distance, parallel processing, data stream

Ευχαριστίες

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας κ. Χρήστο Δουλκερίδη, Αναπληρωτή Καθηγητή στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς, για την πολύτιμη βοήθειά του, τις συμβουλές του και τις παρατηρήσεις του κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας.

Θα ήθελα επίσης να ευχαριστήσω και τον κ. Γεώργιο Σαντιπαντάκη, μεταδιδακτορικό ερευνητή στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς για την βοήθειά του και τις πολύ χρήσιμες υποδείξεις του σχετικά με την επίλυση του προβλήματος που πραγματεύεται η διπλωματική εργασία.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξε σε όλη τη διάρκεια των σπουδών μου και συνεχίζει να με στηρίζει σε κάθε μου βήμα.

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Πίνακας Πινάκων

1 Εισαγωγή

1.1 Πρόλογος

Στη σημερινή εποχή, τα δεδομένα κάθε είδους παίζουν κυρίαρχο ρόλο στην καθημερινή μας ζωή και επηρεάζουν όλες τις δραστηριότητες, από τις πιο απλές, έως τις πιο σύνθετες. Τα μεγάλα δεδομένα είναι παντού και η αποδοτική και γρήγορη ανάλυσή τους, μας βοηθά να βγάζουμε χρήσιμα συμπεράσματα και να βελτιώνουμε τις εφαρμογές μας, τις υπηρεσίες μας και τον τρόπο λήψης αποφάσεων. Οι Sonawane et al. [19], αναφέρουν ότι τα Μεγάλα Δεδομένα χαρακτηρίζονται από τα τρία Vs: *Volume* (*Όγκος*), *Velocity* (*Ταχύτητα*) και *Variety* (*Ποικιλία*) (Εικόνα 1).

Εικόνα 1: Τα τρία χαρακτηριστικά των μεγάλων δεδομένων

Καθημερινά παράγεται ολοένα και αυξανόμενος όγκος δεδομένων, τα οποία βρίσκονται σε αρχεία καταγραφής, εφαρμογές, και συναλλαγές κάθε είδους.

Τα μεγάλα δεδομένα επίσης παράγονται με μεγάλη ταχύτητα, ευρισκόμενα είτε σε σύνολα δεδομένων γνωστά εκ των προτέρων, είτε με τη μορφή ροών δεδομένων.

Επιπλέον, τα μεγάλα δεδομένα έρχονται σε μια ποικιλία μορφών, και μπορεί να έχουν ή να μην έχουν συγκεκριμένη δομή.

Ο ολοένα και αυξανόμενος όγκος και ρυθμός παραγωγής των δεδομένων αυτών, και οι ποικίλες μορφές στις οποίες δημιουργούνται, προϋποθέτουν την ύπαρξη κατάλληλων υποδομών υλικού και λογισμικού, οι οποίες είναι σε θέση να επεξεργαστούν τα δεδομένα με τεχνικές παράλληλης επεξεργασίας, σε ένα σύνολο από κόμβους, εξασφαλίζοντας ταχύτητα και αξιοπιστία στην επεξεργασία.

Η ναυτιλία είναι ένας κλάδος ο οποίος μπορεί να ωφεληθεί στο μέγιστο βαθμό από την ανάλυση των μεγάλων δεδομένων. Καθημερινά, εκτελούνται πολλά δρομολόγια πλοίων και με χρήση ειδικών συσκευών, είμαστε σε θέση να καταγράψουμε και να αποθηκεύσουμε τη θέση, αλλά και την κατάσταση του κάθε πλοίου ανά τακτά χρονικά διαστήματα. Η ανάλυση αυτών των *χωροχρονικών* δεδομένων και ο εντοπισμός των ακολουθούμενων δρομολογίων μπορεί να μας βοηθήσει να εξάγουμε χρήσιμα συμπεράσματα, τα οποία θα μπορούσαν να αφορούν για παράδειγμα, την εύρεση των πιο συχνών χρησιμοποιούμενων θαλάσσιων διαδρομών, η παρακολούθηση του στόλου των πλοίων μιας εταιρείας, και ο αποδοτικός σχεδιασμός του προγράμματος δρομολογίων.

1.2 Σκοπός της διπλωματικής εργασίας

Σε αυτή την εργασία, έχουμε σκοπό να επεξεργαστούμε δεδομένα κίνησης πλοίων σε πραγματικό χρόνο. Συγκεκριμένα, ως είσοδο στον αλγόριθμό μας θα δεχόμαστε μηνύματα που αντιστοιχούν σε θέσεις πλοίων που εκτελούν το ταξίδι τους, και ένα σύνολο από προκαθορισμένες διαδρομές που ορίζονται σε μια συγκεκριμένη γεωγραφική περιοχή. Οι διαδρομές αυτές αντιπροσωπεύουν εναλλακτικές πορείες πλεύσης κατά την είσοδο σε κάποιο λιμάνι, κατά την έξοδο από αυτό ή εναλλακτικές πορείες για την πραγματοποίηση του δρομολογίου μεταξύ δύο λιμανιών. Στόχος μας είναι, για κάθε πλοίο, γνωρίζοντας το λιμάνι αναχώρησης ή/και προορισμού, να εντοπίσουμε σε πραγματικό χρόνο, ποιο δρομολόγιο (διαδρομή) ακολουθείται από το πλοίο, από τα εναλλακτικά δρομολόγια που αφορούν το τρέχον ταξίδι. Το αποτέλεσμα του αλγόριθμου θα περιλαμβάνει, ανά χρονικά διαστήματα, για κάθε πλοίο και ταξίδι, τις υποψήφιες εναλλακτικές διαδρομές που αυτό ακολουθεί, ταξινομημένες με βάση μια συγκεκριμένη μετρική απόστασης, από την πιο πιθανή, έως τη λιγότερο πιθανή.

1.3 Δομή της διπλωματικής εργασίας

Στο 1ο κεφάλαιο, παρουσιάζουμε μια σύντομη εισαγωγή και αναφερόμαστε στο σκοπό της εργασίας.

Στο 2ο κεφάλαιο κάνουμε μια ανασκόπηση του θεωρητικού και τεχνολογικού υποβάθρου, καθώς και αναφορά σε συναφείς εργασίες.

Στο 3ο κεφάλαιο περιγράφουμε τον αλγόριθμο της εφαρμογής μας, τα σύνολα δεδομένων και την προ-επεξεργασία τους, καθώς και την υλοποίηση της εφαρμογής μας.

Στο 4ο κεφάλαιο εκτιμούμε την απόδοση της εφαρμογής, πειραματιζόμενοι με διάφορες παραμέτρους.

Στο 5ο κεφάλαιο παραθέτουμε ενδεικτικές μελέτες περίπτωσης στα δεδομένα της εφαρμογής, με παρουσίαση των αποτελεσμάτων και οπτικοποίηση.

Τέλος, στο 6ο κεφάλαιο παραθέτουμε τα συμπεράσματά μας και προτάσεις για μελλοντικές επεκτάσεις και βελτιώσεις.

2 Θεωρητικό και Τεχνολογικό Υπόβαθρο

2.1 Το σύστημα A.I.S. (Automatic Identification System)

Το σύστημα A.I.S.[4] είναι ένα αυτόματο σύστημα εντοπισμού πλοίων, το οποίο βασίζεται στη λειτουργία πομποδεκτών εγκατεστημένους στα πλοία, οι οποίοι στέλνουν και λαμβάνουν μηνύματα συγκεκριμένων τύπων. Τα μηνύματα περιλαμβάνουν πληροφορίες όπως την τρέχουσα γεωγραφική θέση, ταχύτητα, πορεία και κατάσταση πλοήγησης (εν κινήσει, σταματημένο). Τα μηνύματα λαμβάνονται τόσο από άλλα πλοία, όσο και από επίγειους σταθμούς λήψης σημάτων AIS.

Το σύστημα AIS είναι χρήσιμο για

- την αποφυγή συγκρούσεων μεταξύ πλοίων
- την παρακολούθηση του στόλου των πλοίων μιας εταιρείας/οργανισμού
- περιπτώσεις έρευνας και διάσωσης
- υποβοήθηση κατά την πλοήγηση
- την ασφάλεια κατά την πλοήγηση
- τη διερεύνηση ατυχημάτων

2.2 Συναφείς εργασίες

Στην ενότητα αυτή, θα αναφερθούμε συνοπτικά σε εργασίες που έχουν πραγματοποιηθεί στο πεδίο της επεξεργασίας συνόλων AIS μηνυμάτων για εντοπισμό τροχιάς πλοίου και εξαγωγής συμπερασμάτων.

Οι Fujino et al. [1], ανέπτυξαν ένα σύστημα, το οποίο, μέσω της επεξεργασίας των AIS μηνυμάτων, αρχικά πραγματοποιεί εντοπισμό μοτίβων που υποδεικνύουν την τροχιά που ακολουθεί ένα πλοίο, και στη συνέχεια, δημιουργεί ειδοποιήσεις σε πραγματικό χρόνο, αν ένα πλοίο βρίσκεται εκτός πορείας. Η πορεία του πλοίου εντοπίζεται μέσω των καταγεγραμμένων γεωγραφικών στιγμάτων του, αλλά και μέσω της κατεύθυνσης στην οποία πλέει. Για την ανάπτυξη του συστήματος χρησιμοποιήθηκαν αρχές και αλγόριθμοι μηχανικής μάθησης, αλλά και ιδέες από το πεδίο της επεξεργασίας φυσικής γλώσσας. Το σύστημα δοκιμάστηκε σε δεδομένα AIS που προέρχονται από περιοχή της βορειοδυτικής Γαλλίας.

Οι Shi et al. [2], ανέπτυξαν ένα σύστημα το οποίο μέσω της επεξεργασίας των AIS μηνυμάτων, πραγματοποιεί εντοπισμό μοτίβων που υποδεικνύουν μη ομαλή συμπεριφορά ενός πλοίου. Οι συγγραφείς αρχικά μοντελοποιούν την έννοια της μη ομαλής συμπεριφοράς ενός πλοίου, και στη συνέχεια αναπτύσσουν έναν αλγόριθμο βασισμένο στην ανάλυση γράφων, για να εξάγουν τροχιές καθώς κινείται το πλοίο. Στη συνέχεια, γίνεται εντοπισμός μη ομαλής συμπεριφοράς ως προς το χώρο. Έπειτα, γίνεται εντοπισμός μη ομαλής συμπεριφοράς ενός πλοίου ως προς τη θεματική πληροφορία.

Οι Filipiak et al. [3], στη μελέτη τους, ασχολούνται με τον εντοπισμό δικτύων θαλάσσιας κυκλοφορίας, μέσω της επεξεργασίας μηνυμάτων AIS. Για το σκοπό αυτό, χρησιμοποιούν έναν εξελικτικό αλγόριθμο. Μέσω του αλγόριθμου γίνεται κατασκευή ενός γράφου που αντιπροσωπεύει ένα δίκτυο θαλάσσιας κυκλοφορίας, το οποίο μπορεί να χρησιμοποιηθεί για το σχεδιασμό δρομολογίων. Για την ανακάλυψη των «σημαντικών» σημείων χρησιμοποιούνται τεχνικές διαμέρισης του χώρου. Η υλοποίηση βασίστηκε στην πλατφόρμα επεξεργασίας μεγάλων δεδομένων *Apache Spark.*

Οι Onyango et al. [15], στη μελέτη τους, ασχολούνται με την εξαγωγή διαδρομών θαλάσσιας κυκλοφορίας, μέσω της επεξεργασίας μηνυμάτων AIS, με τη χρήση μη εποπτευόμενου αλγόριθμου μηχανικής μάθησης. Η προσέγγιση αποτελείται από τρία βήματα: Ανακάλυψη σημείων καμπής (*maneuvering points*), ανακάλυψη σημείων (waypoints) διαδρομών και κατασκευή του δικτύου θαλάσσιας κυκλοφορίας. Αρχικά, γίνεται προ-επεξεργασία και καθαρισμός του συνόλου από παρελθοντικά μηνύματα AIS, και στη συνέχεια γίνεται ο εντοπισμός των σημείων στα οποία το πλοίο αλλάζει πορεία. Τέλος, εφαρμόζεται ο αλγόριθμος συσταδοποίησης (*clustering*) HDBSCAN, ο οποίος αποτελεί βελτιωμένη επέκταση του αλγόριθμου DBSCAN. Η μέθοδος της μελέτης δοκιμάστηκε σε δεδομένα AIS που προέρχονται από την περιοχή του λιμανιού της πόλης Ίντσον (Incheon) στη βορειοδυτική Νότια Κορέα.

Οι Dobrkovic et al. [16], στην έρευνά τους, επεξεργάστηκαν μηνύματα AIS με σκοπό τον εντοπισμό μοτίβων θαλάσσιας κυκλοφορίας και τη μετατροπή τους σε ένα κατευθυνόμενο γράφο που μπορεί να χρησιμοποιηθεί για να εντοπιστούν τροχιές και προορισμοί πλοίων. Η ερευνητική ομάδα χρησιμοποιεί έναν γενετικό αλγόριθμο για να πραγματοποιήσει συσταδοποίηση των δεδομένων των θέσεων των πλοίων. Προκειμένου να επιτευχθεί γρηγορότερη επεξεργασία των δεδομένων, χρησιμοποιείται μια δενδρική δομή (*Quad Tree*) για την ευρετηρίαση των δεδομένων. Η μέθοδος δοκιμάστηκε σε δεδομένα AIS που προέρχονται από δύο περιοχές της Ολλανδίας, ενώ έγινε και προσομοίωση με χρήση συνθετικών δεδομένων.

Οι Pallotta et al. [17], ανέπτυξαν ένα εργαλείο για τον αυτόματο εντοπισμό διαδρομών και μοτίβων μη αναμενόμενης συμπεριφοράς σε πλοία, μέσω της επεξεργασίας επίγειων και δορυφορικών AIS μηνυμάτων. Τα αποτελέσματα της επεξεργασίας αποθηκεύονται σε μια δομή δεδομένων που περιέχει δεδομένα πλοίων, σημείων και τροχιών. Για την επεξεργασία των δεδομένων χρησιμοποιείται ο αλγόριθμος συσταδοποίησης DBSCAN.

Οι Sheng et al. [18], στη μελέτη τους, επίσης ασχολούνται με τον αυτόματο εντοπισμό μοτίβων τροχιών που ακολουθούνται από πλοία, μέσω της επεξεργασίας μηνυμάτων AIS με τεχνικές μη εποπτευόμενης μηχανικής μάθησης. Οι ερευνητές αναπτύσσουν ένα μοντέλο συσταδοποίησης σε τέσσερα βήματα: προ-επεξεργασία των δεδομένων, μέτρηση ομοιότητας δομών, συσταδοποίηση και εξαγωγή τροχιών. Ο αλγόριθμος συσταδοποίησης DBSCAN χρησιμοποιήθηκε για την επεξεργασία, ενισχυμένος με μια «Συνθετική Μετρική Απόστασης» (*Synthetic Distance Function*) για τη μέτρηση της ομοιότητας των συστάδων (τροχιών) που προκύπτουν. Τα βήματα της μελέτης εφαρμόστηκαν σε ένα σύνολο δεδομένων AIS από φορτηγά πλοία στην περιοχή του λιμανιού της πόλης Τιεντζίν (Tianjin) της Κίνας.

2.3 Κατανεμημένη Επεξεργασία μεγάλων δεδομένων

Η ανάγκη επεξεργασίας των μεγάλων δεδομένων, πολλές φορές εξαντλεί τα όρια της μνήμης, των συστημάτων αρχείων και της επεξεργαστικής ισχύος ενός μεμονωμένου συστήματος, κι έτσι πλέον έχει κριθεί απαραίτητη η ανάπτυξη εφαρμογών και συστημάτων αρχείων, τα οποία δουλεύουν σε κατανεμημένο περιβάλλον. Αυτό σημαίνει ότι τη φιλοξενία και την επεξεργασία των δεδομένων, αναλαμβάνουν από κοινού περισσότεροι από ένας κόμβοι ενός υπολογιστικού συστήματος, και η επεξεργασία και η αποθήκευση εκτελείται παράλληλα στους κόμβους, εξασφαλίζοντας έτσι ταχύτερη και αποδοτικότερη επεξεργασία. Τέτοια συστήματα είναι επίσης *επεκτάσιμα*, δηλαδή μπορούν να προστεθούν νέοι κόμβοι στο σύστημα εφόσον αυξηθούν οι υπολογιστικές ανάγκες και απαιτηθούν επιπλέον υπολογιστικοί πόροι.

Η επεξεργασία των δεδομένων σε τέτοια συστήματα, γίνεται με δύο μεθόδους, την επεξεργασία σε *τεμάχια* (*batches*) και την επεξεργασία σε *ροές δεδομένων* (*streams*) [5].

Κατά την batch επεξεργασία, γνωρίζουμε εκ των προτέρων το μέγεθος του συνόλου δεδομένων, και μπορούμε να το ανακτήσουμε ολόκληρο, ενώ και η επεξεργασία τελειώνει σε πεπερασμένο χρόνο, όταν ολοκληρωθεί η επεξεργασία του συνόλου δεδομένων.

Αντίθετα, κατά την streaming επεξεργασία, δεν γνωρίζουμε εκ των προτέρων το μέγεθος του συνόλου δεδομένων, και καλούμαστε να επεξεργαστούμε τα δεδομένα σε πραγματικό χρόνο όπως αυτά εισέρχονται στο σύστημα. Η επεξεργασία τους, θεωρητικά διαρκεί για *απεριόριστο* χρόνο, όσο εισέρχονται δεδομένα στο σύστημα. Τα συστήματα επεξεργασίας ροών δεδομένων εφαρμόζουν τεχνικές διαμέρισης, όπως για παράδειγμα αυτές που βασίζονται στο χρόνο, έτσι ώστε να καταφέρουν να επεξεργαστούν το σύνολο δεδομένων χρησιμοποιώντας τους πεπερασμένους υπολογιστικούς πόρους του κάθε κόμβου.

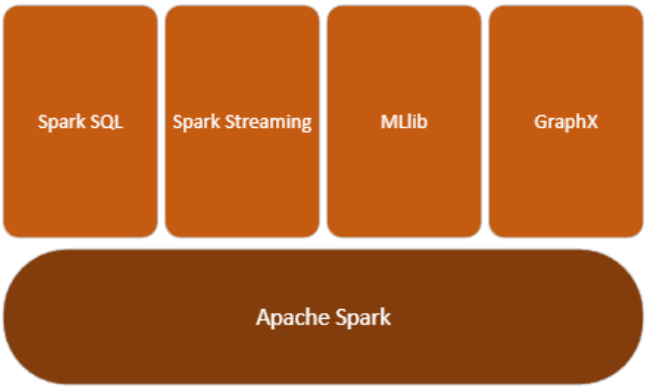
Για τις ανάγκες της επεξεργασίας των δεδομένων της παρούσας εργασίας, αναπτύχθηκε κώδικας σε γλώσσα προγραμματισμού Python, εκτελούμενος στην πλατφόρμα επεξεργασίας μεγάλων δεδομένων *Apache Spark*, και συγκεκριμένα, στη μονάδα *Spark Streaming*.

2.4 Το Apache Spark

2.4.1 Δομικά στοιχεία

Το Apache Spark είναι μια πλατφόρμα επεξεργασίας μεγάλων δεδομένων γενικής χρήσης. Προσφέρει δυνατότητες επεξεργασίας δεδομένων τόσο σε επεξεργασία batch, όσο και streaming. Παρέχει υψηλού επιπέδου διεπαφές (APIs) για τις γλώσσες προγραμματισμού Scala, Java, Python και R.

Το Apache Spark αποτελείται από τα παρακάτω στοιχεία (Εικόνα 2) [5]:



Εικόνα 2: Τα δομικά στοιχεία του Apache Spark [5] [7]

Apache Spark Core (πυρήνας): Περιλαμβάνει τον κύριο μηχανισμό επεξεργασίας δεδομένων και παρέχει προγραμματιστικές διεπαφές (APIs) για την κατανεμημένη επεξεργασία δεδομένων από μια *συστάδα* (cluster) υπολογιστών. Η κύρια δομή δεδομένων που χρησιμοποιείται για την παράλληλη επεξεργασία των δεδομένων στους κόμβους είναι το RDD (resilient Distributed Dataset). Θα μπορούσαμε να πούμε ότι το RDD και η επέκτασή του, το DataFrame, εννοιολογικά μοιάζουν με πίνακα μιας σχεσιακής βάσης δεδομένων ή ένα αρχείο λογιστικού φύλλου, όπου τα δεδομένα οργανώνονται σε «γραμμές» και «στήλες».

Spark SQL: Περιλαμβάνει προγραμματιστικές διεπαφές υψηλού επιπέδου για τις δομές δεδομένων *Dataset* και *DataFrame*, τις οποίες υποστηρίζει το Spark. Παρέχει τη δυνατότητα δημιουργίας ερωτημάτων διατυπωμένων στη γλώσσα επερωτήσεων Structured Query Language (SQL), η οποία χρησιμοποιείται για τη διαχείριση των δεδομένων στις κλασικές, σχεσιακές βάσεις δεδομένων, παρέχοντας έτσι ένα χρήσιμο εργαλείο επεξεργασίας των δεδομένων, το οποίο είναι ήδη ευρέως γνωστό και διαδεδομένο. Μέσω της χρήσης του μηχανισμού βελτιστοποίησης ερωτημάτων Catalyst, το Spark είναι σε θέση να υπολογίσει και εκτελέσει ένα αποδοτικό πλάνο εκτέλεσης για κάθε ερώτημα, εξασφαλίζοντας ταχύτητα στην επεξεργασία και αποδοτικότερη χρήση πόρων.

MLlib: Περιλαμβάνει αλγόριθμους μηχανικής μάθησης, για εργασίες όπως *συσταδοποίηση* (clustering) και ταξινόμηση (classification).

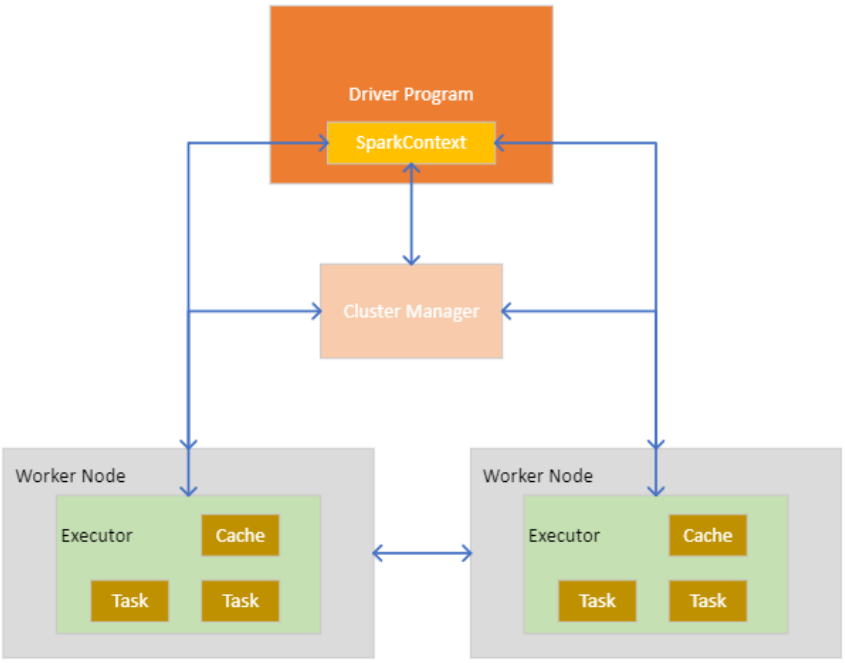
GraphX: Περιλαμβάνει αλγόριθμους για εργασίες ανάλυσης γράφων.

Spark Streaming: περιλαμβάνει προγραμματιστικές διεπαφές για την επεξεργασία δεδομένων σε (σχεδόν) πραγματικό χρόνο. Βασίζεται στην ιδέα της μετατροπής συνεχών ροών δεδομένων σε διακριτές συλλογές δεδομένων, τις οποίες έπειτα μπορούν να επεξεργαστούν οι μηχανισμοί παράλληλης επεξεργασίας του πυρήνα του Spark. Αυτές οι διακριτές συλλογές δεδομένων καλούνται «μικρό-τεμάχια» (micro-batches) και η κύρια δομή δεδομένων που χρησιμοποιείται εδώ είναι η *Διακριτή Ροή* (*Discretized Stream*, *DStream*).

Η επέκταση του Spark Streaming, το *Structured Streaming*, παρέχει τις SQL λειτουργίες που υποστηρίζονται στα DataFrames, και τις επεκτείνει για την επεξεργασία ροών δεδομένων. Είναι το εργαλείο πάνω στο οποίο αναπτύχθηκε η παρούσα διπλωματική εργασία.

2.4.2 Αρχιτεκτονική και τρόπος λειτουργίας

Το Spark βασίζεται σε αρχιτεκτονική «Αφέντη-Σκλάβου» (Master-Slave), η οποία φαίνεται στην παρακάτω εικόνα (Εικόνα 3) [5] [6]:

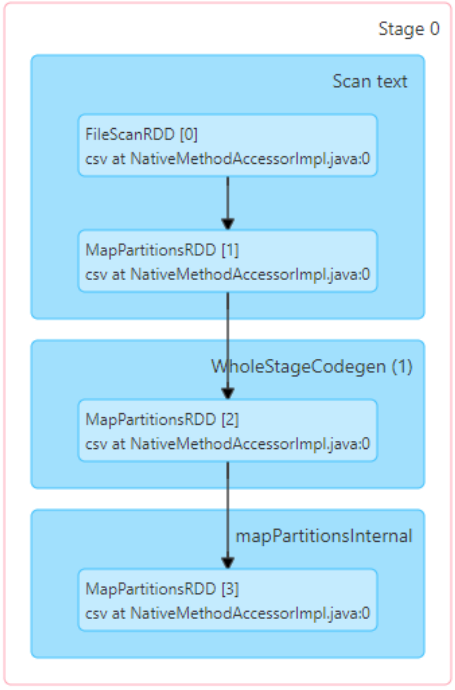


Εικόνα 3: Η αρχιτεκτονική του Apache Spark [7]

Ο κεντρικός (master) κόμβος, εκτελεί το πρόγραμμα-οδηγό (Driver Program), το οποίο είναι και το σημείο εκκίνησης της εφαρμογής. Το Driver Program δημιουργεί το αντικείμενο SparkContext, μέσω του οποίου γίνεται η επικοινωνία με τη διεργασία διαχείρισης συστάδας (Cluster Manager). Το Driver Program διαχωρίζει τον υπολογιστικό φόρτο της εφαρμογής σε μικρότερες μονάδες εκτέλεσης, τις Εργασίες (Tasks). Τα Tasks οποία εκτελούνται από τους Executors, οι οποίοι είναι διεργασίες που εκτελούνται στους δευτερεύοντες κόμβους της συστάδας, τους Workers.

Οι ενέργειες που μπορούν να γίνουν σε ένα RDD (και αντίστοιχα σε DataFrame) μετά τη δημιουργία του είναι δύο ειδών: Οι μετασχηματισμοί (Transformations) και οι Ενέργειες (Actions). Τα RDDs και τα DataFrames δεν μπορούν να τροποποιηθούν μετά τη δημιουργία τους. Είναι “Immutable”.

Έτσι, παράγουμε νέα DataFrames, ορίζοντας κάποιους μετασχηματισμούς πάνω σε αυτά, όπως για παράδειγμα το να δημιουργήσουμε μια νέα στήλη με βάση κάποια συνθήκη ή υπολογισμό. Στο σημείο αυτό, το Spark δεν κάνει κάποια επεξεργασία, αλλά ξεκινά να δουλεύει μόλις ορίσουμε μία Ενέργεια (Action), όπως είναι για παράδειγμα η αποθήκευση ενός DataFrame σε ένα κατανεμημένο σύστημα αρχείων. Το Spark καταγράφει τις εντολές (μετασχηματισμούς) που πρέπει να εκτελεστούν, σε ένα διάγραμμα ροής, το DAG (Directed Acyclic Graph), έως ότου έρθει η κατάλληλη στιγμή να τις εκτελέσει, ως αποτέλεσμα κάποιας ενέργειας. Αυτός ο τρόπος εκτέλεσης (“Lazy Evaluation”) έχει το πλεονέκτημα ότι το Spark μπορεί να βελτιστοποιήσει το πλάνο εκτέλεσης και να διαχειριστεί αποδοτικά τους διαθέσιμους υπολογιστικούς πόρους της συστάδας. Στην Εικόνα 4 φαίνεται ένα τέτοιο DAG:



Εικόνα 4: Το DAG (Directed Acyclic Graph)

Τα δεδομένα καθενός DataFrame, χωρίζονται εσωτερικά από το Spark σε υποσύνολα, τα οποία λέγονται *διαμερίσεις* (*partitions*) [6]. Η κάθε διαμέριση, είναι ένα υποσύνολο από τα δεδομένα («γραμμές»), τα οποία βρίσκονται στη μνήμη ενός κόμβου της συστάδας. Τα partitions των δεδομένων διανέμονται στους workers προς επεξεργασία κι έτσι το Spark είναι σε θέση να εκτελέσει τους υπολογισμούς παράλληλα.

2.4.3 Spark Streaming

2.4.3.1 Μοντέλο Επεξεργασίας

Γενικά, τα δομικά στοιχεία της διαδικασίας που ακολουθείται στα συστήματα επεξεργασίας ροών δεδομένων (άρα και στο Spark Streaming) είναι τρία (Εικόνα 5) [5]:

- Πηγές (sources) δεδομένων, που μπορεί να είναι στίγματα θέσεων πλοίων, μετρήσεις αισθητήρων, χρηματοοικονομικές συναλλαγές
- Επεξεργασία των ροών δεδομένων
- Έξοδοι (sinks) δεδομένων που μπορεί να είναι κατανεμημένα συστήματα αρχείων, η κονσόλα του κόμβου που εκτελεί το Driver Program, κινητές συσκευές-πελάτες (clients) κλπ.

Εικόνα 5: Τα στάδια επεξεργασίας ροών δεδομένων [5]

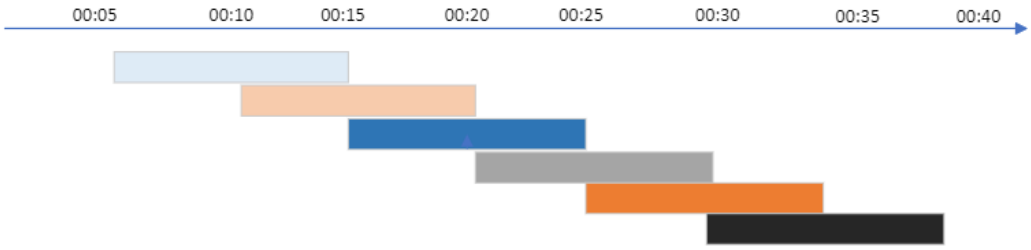
2.4.3.2 Συναθροίσεις με βάση το χρόνο

Στο streaming processing, όπως και στο batch processing, ορίζουμε μετασχηματισμούς (Transformations) που δέχονται ως είσοδο μια ροή δεδομένων (DStream ή Streaming DataFrame) και παράγουν μια άλλη ροή. Πολύ σημαντική είναι και η έννοια των Συναθροίσεων (Aggregations). Κάθε μετασχηματισμός μετασχηματίζει ένα προς ένα το κάθε στοιχείο της συλλογής και παράγει ένα νέο. Έτσι, για παράδειγμα, ένας μετασχηματισμός που υψώνει στο τετράγωνο μια αριθμητική τιμή, βασίζεται μόνο σε αυτή την τιμή για να παράγει το νέο αποτέλεσμα. Αντίθετα, οι συναθροίσεις, παράγουν αποτελέσματα, τα οποία βασίζονται στην επεξεργασία περισσότερων από ένα στοιχείων της συλλογής. Για παράδειγμα, ο υπολογισμός του αριθμητικού μέσου είναι μια συνάθροιση. Ένα πολύ χρήσιμο είδος συνάθροισης που μπορούμε να κάνουμε με το spark Streaming είναι οι συναθροίσεις με βάση το χρόνο ή αλλιώς, «Συναθροίσεις Χρονικού Παράθυρου» (*Window Aggregations*) [5]. Μια συνάθροιση χρονικού παράθυρου θα μπορούσε να είναι να υπολογίσουμε το πλήθος των συμβάντων σφάλματος στο χρονικό διάστημα των τελευταίων 15 λεπτών.

"Tumbling Window": Με την τεχνική αυτή, υπολογίζουμε μια συνάθροιση ανά κάποιο χρονικό διάστημα, για παράδειγμα να υπολογίσουμε τη μέση ταχύτητα ενός πλοίου λαμβάνοντας υπόψη τα AIS μηνύματά του ανά 10 λεπτά. Οι χρονικές περίοδοι δεν επικαλύπτονται. Κάθε ομάδα (group) στα δεδομένα, υπολογίζεται ανά τη μονάδα χρόνου που ορίζουμε και ακολουθεί την προηγούμενη, χωρίς επικάλυψη, ενώ είναι και ανεξάρτητη από εκείνη. Στην επόμενη εικόνα (Εικόνα 6) φαίνεται η συνάθροιση "tumbling window":

Εικόνα 6: Tumbling Window

"Sliding Window": Με την τεχνική αυτή, υπολογίζουμε συναθροίσεις ανά χρονικά διαστήματα, όμως ανανεώνουμε τον υπολογισμό τους συχνότερα από όσο ορίζει το χρονικό διάστημα συνάθροισης. Μια τέτοια συνάθροιση θα μπορούσε να είναι ο υπολογισμός της μέσης ημερήσιας θερμοκρασίας που καταγράφει ένας αισθητήρας, αλλά να δείχνουμε την τιμή αυτή ανά ώρα. Είναι δηλαδή ένας κυλιόμενος αριθμητικός μέσος. Η επόμενη εικόνα (Εικόνα 7) αναπαριστά μια συνάθροιση τύπου Sliding Window:



Εικόνα 7: Sliding Window

2.4.3.3 "Stateless" και "Stateful" Processing

Μία επίσης πολύ σημαντική έννοια που πρέπει να έχουμε υπόψη μας σχεδιάζοντας μια εφαρμογή επεξεργασίας ροών δεδομένων, είναι η έννοια της «Κατάστασης» ("State") [5]. Σε μια stateless εφαρμογή επεξεργασίας ροών δεδομένων, η επεξεργασία του κάθε «τεμαχίου» (batch) δεδομένων γίνεται χωρίς να λάβουμε υπόψη τα προηγούμενά του. Σε μια stateful εφαρμογή, για να κατανοήσουμε τι συμβαίνει με τα δεδομένα που λάβαμε τώρα, θα πρέπει να λάβουμε υπόψη και δεδομένα που είχαμε λάβει παλιότερα. Για παράδειγμα, αν σε μια εφαρμογή που λαμβάνει συμβάντα ενός server σε πραγματικό χρόνο μας ενδιαφέρει να στέλνουμε μια προειδοποίηση κάθε φορά που λαμβάνουμε ένα συμβάν σφάλματος, τότε αυτή η εφαρμογή θα αδιαφορεί για το τι συνέβη στα προηγούμενα συμβάντα κι έτσι θα είναι "stateless".

Αντίθετα, αν σε αυτή την εφαρμογή, μας ενδιαφέρει να μετράμε το πλήθος των σφαλμάτων που συμβαίνουν κάθε 10 λεπτά, τότε για να γίνει ο υπολογισμός, η εφαρμογή θα πρέπει να λαμβάνει υπόψη, εκτός από τη χρονική σήμανση των συμβάντων του τρέχοντος batch, και όλα τα συμβάντα με χρονική σήμανση εντός του τελευταίου 10λέπτου, τα οποία πιθανόν να είχαν ληφθεί σε προηγούμενα batches. Έτσι, η εφαρμογή μας είναι τώρα "Stateful".

Όταν έχουμε stateful εφαρμογή, είναι χρήσιμο να έχουμε ορίσει και το υδατογράφημα (*watermark*). Το υδατογράφημα εκφράζει χρονική διάρκεια. Τα δεδομένα τα οποία έχουν ηλικία (σύμφωνα με το χρόνο των γεγονότων) μεγαλύτερη από το watermark, θα αγνοούνται από το σύστημα, είτε έχουν έρθει «στην ώρα τους», είτε με καθυστέρηση. Αν για παράδειγμα έχουμε ορίσει ως watermark τα 10 λεπτά, συμβάντα ηλικίας μεγαλύτερης των 10 λεπτών θα απορρίπτονται και δεν θα λαμβάνονται πλέον υπόψη για τους υπολογισμούς.

Το Spark Streaming υποστηρίζει τόσο Stateful, όσο και Stateless επεξεργασία. Η χρήση των Window Aggregations στην εφαρμογή μας (όπως αναφέρθηκαν στην ενότητα 2.4.3.2), είναι ένας τρόπος να επιτύχουμε Stateful σχεδιασμό.

3 Περιγραφή Υλοποίησης

3.1 Περιγραφή του αλγόριθμου

Στην ενότητα αυτή, θα περιγράψουμε τον αλγόριθμου που υλοποιείται μέσω της εφαρμογής που αναπτύξαμε. Ο αλγόριθμος έχει ως στόχο τον εντοπισμό του δρομολογίου που είναι περισσότερο πιθανό ένα πλοίο να ακολουθεί καθώς ταξιδεύει από έναν προορισμό σε έναν άλλο. Η εφαρμογή μας θα λαμβάνει ως είσοδο τροχιές κινούμενων πλοίων και δρομολόγια (διαδρομές).

Η *τροχιά* του πλοίου είναι μια ακολουθία σημείων $(X_1,Y_1)_{t1}$, $(X_2,Y_2)_{t2}$,... $(X_n,Y_n)_{tn}$, όπου το κάθε σημείο προσδιορίζεται από το γεωγραφικό μήκος (longitude) X_i και το γεωγραφικό πλάτος (latitude) Y_i της θέσης του πλοίου, όπως αποτυπώνεται τη χρονική στιγμή t_i.

Το *δρομολόγιο* (*διαδρομή*) είναι μια ακολουθία σημείων (X_1,Y_1), (X_2,Y_2),... (X_n,Y_n), όπου το κάθε σημείο προσδιορίζεται από το γεωγραφικό μήκος (longitude) X_i και το γεωγραφικό πλάτος (latitude) Y_i της θέσης του. Ανάμεσα σε δύο διαδοχικά σημεία της διαδρομής, περνά μια νοητή ευθεία γραμμή κι έτσι τελικά η κάθε διαδρομή λαμβάνεται υπόψη ως ένα σύνολο ευθυγράμμων τμημάτων. Αν μια διαδρομή περιλαμβάνει n σημεία, θα αποτελείται από n-1 ευθύγραμμα τμήματα.

Η τροχιά του κάθε πλοίου, γίνεται γνωστή μέσω μηνυμάτων A.I.S που εκπέμπει το πλοίο καθώς εκτελεί το ταξίδι του. Το κάθε μήνυμα περιλαμβάνει, μεταξύ άλλων, το μοναδικό αναγνωριστικό του πλοίου, την τρέχουσα θέση του, καθώς και το λιμάνι αναχώρησης, το λιμάνι προορισμού, ή και τα δύο.

Τα εναλλακτικά δρομολόγια είναι γνωστά εκ των προτέρων. Κάθε εναλλακτικό δρομολόγιο αφορά μια πορεία πλεύσης για την είσοδο ή την έξοδο από ένα λιμάνι, είτε την πορεία πλεύσης μεταξύ δύο λιμανιών.

Ο έγκαιρος εντοπισμός του πιθανότερου δρομολογίου που ακολουθεί ένα πλοίο, έχει μεγάλη χρησιμότητα και προσφέρει πλεονεκτήματα για το διαχειριστή του στόλου των πλοίων μιας εταιρείας, ακόμα και για τις λιμενικές αρχές, διότι:

- Μπορεί να γίνει ευκολότερα η πρόβλεψη για την ώρα άφιξης του πλοίου στον προορισμό,
- Συμβάλει στον καλύτερο προγραμματισμό των δρομολογίων
- Μπορούν να εντοπιστούν συγκεκριμένες περιοχές και θαλάσσιες οδοί που συγκεντρώνουν μεγαλύτερη και μικρότερη κίνηση πλοίων. Έτσι, οι εναλλακτικές διαδρομές και ο έλεγχος κυκλοφορίας μπορούν να βελτιωθούν
- Είναι ευκολότερο να εντοπιστούν ποια πλοία βρίσκονται κοντά σε περιοχές ενδιαφέροντος, όπως λιμάνια με πολλή κίνηση, απαγορευμένες περιοχές, κλπ.

Ο αλγόριθμος με τον οποίο θα επιλύσουμε το πρόβλημα του εντοπισμού δρομολογίου, έχει ως εξής:

1. Φόρτωση των εναλλακτικών δρομολογίων. Για όλα τα σημεία του κάθε δρομολογίου, δημιουργούμε ευθύγραμμα τμήματα μεταξύ δύο διαδοχικών σημείων, δηλαδή τα χωρίζουμε σε ζευγάρια των δύο σημείων, διατηρώντας τη σωστή ταξινόμηση.
2. Εκκίνηση της λήψης της ροής των A.I.S. μηνυμάτων με τις θέσεις των πλοίων
3. Για κάθε μήνυμα που λαμβάνουμε:

a. Εύρεση των εναλλακτικών διαδρομών που αφορούν το συγκεκριμένο ταξίδι του πλοίου (με βάση τους λιμένες αναχώρησης και προορισμού)
b. Υπολογισμός της απόστασης της τρέχουσας θέσης του πλοίου από όλα τα ευθύγραμμα τμήματα που ορίζονται από τα σημεία του κάθε δρομολογίου. Για τον υπολογισμό αυτής της απόστασης, χρησιμοποιούμε τη μετρική της απόστασης σημείου από ευθύγραμμο τμήμα, που ορίζεται από τον τύπο (1) παρακάτω.

4. Ανά τακτά χρονικά διαστήματα, και για κάθε πλοίο, απόδοση βαθμολογίας σε κάθε εναλλακτικό δρομολόγιο που αφορά το συγκεκριμένο ταξίδι, με χρήση συγκεκριμένης μετρικής. Η μετρική που θα χρησιμοποιήσουμε εμείς για την απόδοση βαθμολογίας (*Score*) σε δρομολόγια, είναι η διάμεσος των αποστάσεων του πλοίου από τα ευθύγραμμα τμήματα της διαδρομής. Έχοντας υπολογίσει την απόσταση της κάθε θέσης του πλοίου από όλα τα ευθύγραμμα τμήματα των εναλλακτικών διαδρομών που αφορούν το συγκεκριμένο ταξίδι, βαθμολογούμε το κάθε δρομολόγιο υπολογίζοντας τη διάμεσο των υπολογισθέντων αποστάσεων για το συγκεκριμένο χρονικό διάστημα, πλοίο, ταξίδι και διαδρομή. Διαδρομές για τις οποίες το πλοίο απέχει μικρότερη διάμεση απόσταση, είναι πιθανότερο να ακολουθούνται.
5. Παρουσίαση των αποτελεσμάτων (είτε στην κονσόλα, είτε αποθήκευση σε αρχείο)

Για τη μέτρηση της απόστασης του σημείου (θέσης) του πλοίου από ένα ευθύγραμμα τμήμα (σκέλος μιας διαδρομής), χρησιμοποιούμε τη μετρική, που ορίζεται από τον τύπο (1) [8]:

$$(1)\ \mathrm{dist}\left(P_1, P_2, (x_0, y_0)\right) = \frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

Έστω (x_0, y_0) οι συντεταγμένες της γεωγραφικής θέσης ενός πλοίου, και $P_1 = (x_1, y_1)$ και $P_2 = (x_2, y_2)$, οι συντεταγμένες των σημείων ενός ευθύγραμμου τμήματος (σκέλους) μιας διαδρομής. Η απόσταση $\mathrm{dist}\left(P_1, P_2, (x_0, y_0)\right)$ της θέσης του πλοίου από την ευθεία που περνάει από τα δύο σημεία, υπολογίζεται ως ο λόγος του διπλάσιου εμβαδού του τριγώνου με κορυφές τα τρία σημεία, προς την Ευκλείδεια απόσταση μεταξύ των δύο σημείων P_1 και P_2.

3.2 Περιγραφή του συνόλου δεδομένων

Τα δεδομένα που χρησιμοποιούμε στη μελέτη μας, αφορούν τη γεωγραφική περιοχή της Νορβηγίας και παρέχονται από την Νορβηγική Διοίκηση Ακτών (Norwegian Coastal Administration (NCA)). Αποτελούνται από:

- Ένα σύνολο από δρομολόγια (*reference routes*) [9] και
- Ένα σύνολο από μηνύματα AIS που αφορούν κινήσεις πλοίων στην περιοχή του Όσλο της Νορβηγίας [10].

3.2.1 Διαδρομές

Παρέχονται συνολικά 528 διαδρομές (πορείες), που καλύπτουν 6 περιοχές της Νορβηγίας (*Oslofjorden, Skagerrak, Rogaland, Vestlandet, More_og_Trondelag* και *Nordland*). Οι 28 από

αυτές συμπεριλαμβάνουν εναλλακτικές διαδρομές. Από αυτές, εμείς θα ασχοληθούμε με τις 16 διαδρομές, που αφορούν την περιοχή του Όσλο, η οποία παρουσιάζει μεγαλύτερο ενδιαφέρον. Οι διαδρομές αποτελούν πορείες πλεύσης κατά την είσοδο ή την έξοδο του πλοίου από συνολικά τέσσερα λιμάνια, στην περιοχή μελέτης μας, τα οποία είναι τα: *Oslo*, *Moss*, *Horten* και *Engene*. Για κάθε λιμάνι έχουμε δύο εναλλακτικές διαδρομές εισόδου και δύο εξόδου, άρα συνολικά τέσσερις διαδρομές.

Ιδιαίτερα για τα λιμάνια Moss και Horten, λόγω του ότι το δρομολόγιο μεταξύ τους είναι πολύ συχνό, και για να επιβεβαιώσουμε την ορθότητα του αλγόριθμου, έχουμε ορίσει τέσσερις επιπλέον διαδρομές, δύο με κατεύθυνση από Moss προς Horten και δύο με κατεύθυνση από Horten προς Moss.

Έτσι, συνολικά το σύνολο δεδομένων μας περιέχει 20 διαδρομές. Κάθε διαδρομή είναι ένα σύνολο σημείων (*waypoints*) με το καθένα να προσδιορίζεται μοναδικά από ένα μοναδικό αναγνωριστικό. Στον επόμενο πίνακα (Πίνακας 1) παραθέτουμε τις διαδρομές που χρησιμοποιούμε στην εφαρμογή μας:

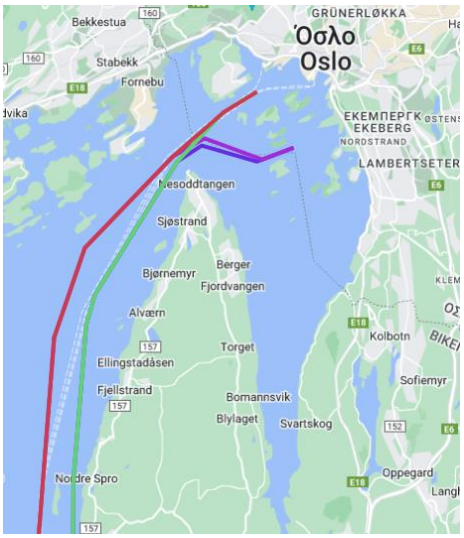
Αρχικό Όνομα Διαδρομής	Σύντομο Όνομα Διαδρομής	Λιμάνι Αναχώρησης	Λιμάνι Προορισμού	Αριθμός Σημείων	Χρώμα στο χάρτη
NCA_EngeneNorth_In_20201001	**EngeneNorth_In**	-	Engene	12	
NCA_EngeneNorth_Out_20201001	**EngeneNorth_Out**	Engene	-	13	
NCA_EngeneSouth_In_20201001	**EngeneSouth_In**	-	Engene	13	
NCA_EngeneSouth_Out_20210212	**EngeneSouth_Out**	Engene	-	13	
NCA_Horten_In_20201001	**Horten_In**	-	Horten	3	
-	**Horten_Moss_North**	Horten	Moss	5	
-	**Horten_Moss_South**	Horten	Moss	5	
NCA_Horten_Out_20201001	**Horten_Out**	Horten	-	3	
NCA_HortenInner_In_20201001	**HortenInner_In**	-	Horten	7	
NCA_HortenInner_Out_20201001	**HortenInner_Out**	Horten	-	6	
-	**Moss_Horten_North**	Moss	Horten	5	
-	**Moss_Horten_South**	Moss	Horten	5	
NCA_MossNorth_In_20201001	**MossNorth_In**	-	Moss	9	
NCA_MossNorth_Out_20210212	**MossNorth_Out**	Moss	-	9	
NCA_MossSouth_In_20201001	**MossSouth_In**	-	Moss	5	

NCA_MossSouth_Out_20201001	**MossSouth_Out**	Moss	-	4	
NCA_OsloEast_In_20201001	**OsloEast_In**	-	Oslo	17	
NCA_OsloEast_Out_20201001	**OsloEast_Out**	Oslo	-	16	
NCA_OsloWest_In_20201001	**OsloWest_In**	-	Oslo	16	
NCA_OsloWest_Out_20210212	**OsloWest_Out**	Oslo	-	15	

Πίνακας 1: Οι διαδρομές που χρησιμοποιούμε στην εφαρμογή μας

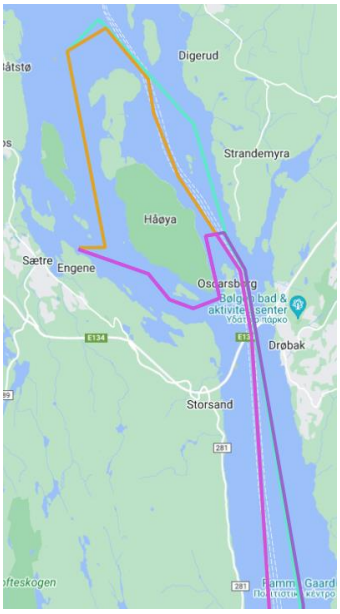
Η στήλη «Αρχικό Όνομα Διαδρομής» υποδηλώνει το αρχικό όνομα της διαδρομής όπως παρέχεται στο σύνολο δεδομένων. Για λόγους καθαρότητας, δώσαμε ένα συντομότερο όνομα στην κάθε διαδρομή (στήλη «Σύντομο Όνομα Διαδρομής»). Τέλος, η στήλη «Χρώμα στο χάρτη», υποδηλώνει το χρώμα της κάθε διαδρομής, όπως φαίνεται στις παρακάτω εικόνες με χάρτη, αλλά και στις εικόνες με χάρτη που παραθέτουμε στο Κεφάλαιο 5.

Στην επόμενη εικόνα (Εικόνα 8), παρατηρούμε τις διαδρομές εισόδου και εξόδου για το λιμάνι Oslo (διαδρομές εισόδου: OsloEast_In, OsloWest_In, διαδρομές εξόδου: OsloEast_Out, OsloWest_Out).



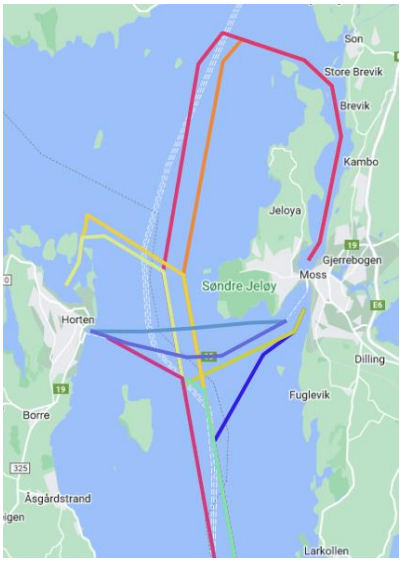
Εικόνα 8: Εναλλακτικές Διαδρομές λιμανιού Oslo

Στην επόμενη εικόνα (Εικόνα 9), παρατηρούμε τις διαδρομές εισόδου και εξόδου για το λιμάνι Engene (διαδρομές εισόδου: EngeneNorth_In, EngeneSouth_In, διαδρομές εξόδου: EngeneNorth_Out, EngeneSouth_Out).



Εικόνα 9: Εναλλακτικές Διαδρομές λιμανιού Engene

Στην επόμενη εικόνα (Εικόνα 10), παρατηρούμε τις διαδρομές για τα λιμάνια Moss και Horten (διαδρομές εισόδου: Horten_In, HortenInner_In, MossNorth_In, MossSouth_In, διαδρομές εξόδου: Horten_Out, HortenInner_Out, MossNorth_Out, MossSouth_Out, διαδρομές μεταξύ των δύο λιμανιών: Horten_Moss_North, Horten_Moss_South, Moss_Horten_North, Moss_Horten_South).



Εικόνα 10: Εναλλακτικές Διαδρομές λιμανιών Horten και Moss

3.2.2 Μηνύματα AIS

Το σύνολο των μηνυμάτων AIS, περιέχει συνολικά 1.420.687 μηνύματα AIS, τα οποία έχουν σταλεί από συνολικά 128 πλοία σε χρονικό διάστημα 10 ημερών (1 έως 10 Σεπτεμβρίου 2022), και καλύπτουν την περιοχή του Όσλο. Τα μηνύματα αυτά περιέχονται σε 10 αρχεία μορφής CSV (ένα για κάθε ημέρα της μελέτης μας), και το καθένα περιέχει τα παρακάτω πεδία (Πίνακας 2):

Όνομα Πεδίου	Χρήση
mmsi	Αναγνωριστικό του πλοίου για σκοπούς επικοινωνίας (*Maritime Mobile Service Identity*)
imo_nr	Αναγνωριστικό του πλοίου κατά IMO (*International Maritime Organization*)
v_length	Μήκος του πλοίου σε μέτρα
date_time_utc	Χρονική σήμανση του μηνύματος στο πρότυπο ώρας UTC
lon	Γεωγραφικό μήκος (longitude) σε δεκαδικές μοίρες
lat	Γεωγραφικό πλάτος (latitude) σε δεκαδικές μοίρες
sog	Ταχύτητα Εδάφους (S*peed Over Ground*), σε κόμβους
cog	Πορεία Εδάφους (*Course Over Ground*), σε δεκαδικές μοίρες
true_heading	Πορεία με βάση τον πραγματικό Βορρά, σε δεκαδικές μοίρες
nav_status	Κατάσταση πλοήγησης (κωδικός που υποδηλώνει την κατάσταση πλοήγησης του πλοίου: εν κινήσει, σταματημένο, κλπ.)
message_nr	Τύπος μηνύματος (κωδικός που υποδηλώνει το είδος του μηνύματος AIS: αναφορά θέσης, αναφορά σταθμού εδάφους, κλπ.)

Πίνακας 2: Πεδία του κάθε AIS μηνύματος στο σύνολο δεδομένων

Με έντονη γραφή επισημαίνονται τα πεδία που θα χρησιμοποιήσουμε στην εφαρμογή μας, δηλαδή το πεδίο του μοναδικού αναγνωριστικού του πλοίου (mmsi), η χρονική στιγμή του μηνύματος (date_time_utc), κα η τρέχουσα θέση του πλοίου (lon, lat).

Στην επόμενη εικόνα (Εικόνα 11) παρατηρούμε μια οπτική απεικόνιση όλων των μηνυμάτων του συνόλου δεδομένων μας:

Εικόνα 11: Το σύνολο δεδομένων εντός της περιοχής μελέτης

3.3 Προ-επεξεργασία του συνόλου δεδομένων

Η προ-επεξεργασία του συνόλου δεδομένων, έχει ως στόχο να το καταστήσει έτοιμο για εισαγωγή στην εφαρμογή μας.

3.3.1 Προ-επεξεργασία των διαδρομών

Ως προς τις διαδρομές, η προ-επεξεργασία περιλαμβάνει τα ακόλουθα βήματα:

- Επιλογή των διαδρομών που αφορούν την περιοχή ενδιαφέροντος. Τα σημεία κάθε διαδρομής περιέχονται σε ένα αρχείο με κωδικοποίηση XML.
- Άνοιγμα του κάθε αρχείου διαδρομής και διατήρηση των πεδίων και των ιδιοτήτων που μας είναι απαραίτητα (συντεταγμένες και όνομα σημείου). Επίσης δημιουργούμε ένα μοναδικό αναγνωριστικό του κάθε σημείου μέσα στη διαδρομή.
- Συγχώνευση όλων των σημείων σε ένα αρχείο, με δημιουργία ενός μοναδικού αναγνωριστικού για κάθε σημείο, αλλά και για κάθε διαδρομή.

Μετά την προ-επεξεργασία αυτή, προκύπτουν δύο αρχεία, το αρχείο “*routes.csv*”, το οποίο περιέχει τα λεκτικά και τα αναγνωριστικά των διαδρομών (Πίνακας 3), και το αρχείο “*waypoints.csv*”, το οποίο περιέχει τα σημεία της κάθε διαδρομής (Πίνακας 4).

Όνομα Πεδίου	Χρήση
route_id	Μοναδικό αναγνωριστικό της διαδρομής
route_type	Τύπος της διαδρομής (1: είναι γνωστό μόνο το λιμάνι αναχώρησης ή προορισμού, 2: είναι γνωστά και τα δύο λιμάνια)

route_name	Σύντομο όνομα της διαδρομής
dep_port	Λιμάνι αναχώρησης
arr_port	Λιμάνι προορισμού

Πίνακας 3: Πεδία του αρχείου "routes.csv" (διαδρομές) μετά την προ-επεξεργασία

Όνομα Πεδίου	Χρήση
wp_id	Μοναδικό αναγνωριστικό του σημείου
route_id	Μοναδικό αναγνωριστικό της διαδρομής στην οποία περιέχεται το σημείο
rwp_id	Αριθμός του σημείου εντός της διαδρομής στην οποία περιέχεται
wp_name	Όνομα του σημείου
lon	Γεωγραφικό μήκος του σημείου, σε δεκαδικές μοίρες
lat	Γεωγραφικό πλάτος του σημείου, σε δεκαδικές μοίρες

Πίνακας 4: Πεδία του αρχείου "waypoints.csv" (σημεία διαδρομών) μετά την προ-επεξεργασία

3.3.2 Προ-επεξεργασία των μηνυμάτων AIS

Ως προς τα μηνύματα AIS, η προ-επεξεργασία περιλαμβάνει τα εξής βήματα:

Συγχώνευση των επιμέρους αρχείων σε ένα και αφαίρεση διπλότυπων: Η υπηρεσία που παρέχει τα δεδομένα, παρέχει ένα csv αρχείο με τα AIS μηνύματα για την κάθε μέρας της περιόδου.

Εντοπισμός τρέχοντος λιμανιού: Είναι σημαντικό να γνωρίζουμε αν, κατά τη χρονική στιγμή ενός μηνύματος, το πλοίο βρίσκεται μέσα σε κάποιο λιμάνι, ώστε να μην υπολογίσουμε αποστάσεις από διαδρομές κι έτσι να εξοικονομήσουμε υπολογισμούς. Επειδή τα μηνύματα δεν περιέχουν πληροφορίες για το τρέχον λιμάνι όπου βρίσκεται το κάθε πλοίο, έπρεπε να τα εισάγουμε εμείς στα μηνύματα. Για να το κάνουμε αυτό, χρησιμοποιήσαμε τη βάση δεδομένων λιμανιών *World Port Index* [11], η οποία περιέχει τις συντεταγμένες του κάθε λιμανιού. Στην Εικόνα 12 παρατηρούμε τη θέση του λιμανιού του Όσλο, όπως έχει καταχωρηθεί στο WPI:

Εικόνα 12: Θέση του λιμανιού Oslo στο W.P.I.

Παρατηρούμε ότι η καταχωρημένη θέση του λιμανιού παρουσιάζει μετατόπιση σε σχέση με το σημείο της προβλήτας, όπου θα αγκυροβολούν τα πλοία. Παράλληλα, κάποια πλοία μπορεί να μην αγκυροβολούν στην προβλήτα, αλλά να παραμένουν λίγο πιο μακριά από αυτή.

Αυτή η μετατόπιση στις θέσεις των λιμανιών παρατηρείται και στα άλλα λιμάνια της περιοχής μελέτης. Έτσι, είναι εμφανές ότι για να θεωρήσουμε ότι ένα πλοίο βρίσκεται σε ένα λιμάνι, θα πρέπει να ορίσουμε μια περιοχή γύρω από αυτό, και κάθε μήνυμα που εκπέμπεται εντός της περιοχής αυτής (κύκλος με συγκεκριμένη ακτίνα), θα θεωρείται ότι εκπέμφθηκε ενώ το πλοίο βρίσκεται μέσα στο λιμάνι.

Έπειτα από δοκιμές και πειραματισμό, ορίσαμε τα 2 χιλιόμετρα ως την ακτίνα του κύκλου γύρω από κάθε λιμάνι, ενώ ορίσαμε και διαφορετικές συντεταγμένες για το κάθε λιμάνι, ώστε να εντοπίζουμε καλύτερα τα πλοία που βρίσκονται εντός της περιοχής αυτής. Στην Εικόνα 13 παρατηρούμε το λιμάνι του Όσλο έχοντας ορίσει νέες συντεταγμένες (κέντρο του κύκλου) και ακτίνα 2 χιλιόμετρα:

Εικόνα 13: Περιοχή του λιμανιού Oslo

Αντίστοιχες παραδοχές πραγματοποιήσαμε και στα υπόλοιπα λιμάνια της περιοχής μελέτης μας. Στη συνέχεια, εκτελέσαμε το χωρικό ερώτημα για να βρούμε τα μηνύματα που βρίσκονται εντός του κάθε κύκλου, και καταχωρήσαμε σε αυτά το όνομα του λιμανιού.

Εντοπισμός λιμανιών αναχώρησης και προορισμού: Έχοντας εισάγει το τρέχον λιμάνι, το επόμενο βήμα είναι να κάνουμε μια εκτίμηση για το λιμάνι αναχώρησης και προορισμού, το οποίο επίσης δεν υπάρχει στα μηνύματα. Είναι όμως σημαντικό να το γνωρίζουμε, ώστε να εξοικονομούμε υπολογισμούς συγκρίνοντας κάθε μήνυμα μόνο με τις διαδρομές που έχουν σχεδιαστεί για το συγκεκριμένο δρομολόγιο.

Έτσι, γνωρίζοντας ότι τη χρονική στιγμή t1 το πλοίο βγαίνει από το λιμάνι Α και τη χρονική στιγμή t2, το πλοίο μπαίνει στο λιμάνι Β, μπορούμε να καταχωρίσουμε στα ενδιάμεσα μηνύματα το λιμάνι αναχώρησης και προορισμού.

Φυσικά, ο αλγόριθμος αυτός, και λόγω της φύσης των δεδομένων (χρονικά διαστήματα χωρίς μηνύματα, αναχώρηση του πλοίου εκτός της περιοχής ενδιαφέροντος, αναχώρηση και επιστροφή στο ίδιο λιμάνι κλπ.), δεν εξασφαλίζει απόλυτη ακρίβεια στον εντοπισμό των λιμανιών αναχώρησης και προορισμού, όμως μας παρέχει ένα ικανοποιητικό ποσοστό ακρίβειας, ώστε να έχουμε ένα σύνολο δεδομένων που να μπορεί να εφαρμοστεί στο πρόβλημά μας.

Επομένως, το τελικά αρχείο csv με τα AIS μηνύματα, περιλαμβάνει τα υπάρχοντα πεδία του κάθε μηνύματος συν τα πεδία του τρέχοντος λιμανιού και των λιμανιών αναχώρησης και προορισμού. Τα πεδία αυτού του αρχείου παρατίθενται στον παρακάτω πίνακα (Πίνακας 5):

Όνομα Πεδίου	Χρήση
mmsi	Αναγνωριστικό του πλοίου για σκοπούς επικοινωνίας (*Maritime Mobile Service Identity*)
imo_nr	Αναγνωριστικό του πλοίου κατά IMO (*International Maritime Organization*)
v_length	Μήκος του πλοίου σε μέτρα
date_time_utc	Χρονική σήμανση του μηνύματος στο πρότυπο ώρας UTC
lon	Γεωγραφικό μήκος (longitude) σε δεκαδικές μοίρες
lat	Γεωγραφικό πλάτος (latitude) σε δεκαδικές μοίρες
sog	Ταχύτητα Εδάφους (S*peed Over Ground*), σε κόμβους
cog	Πορεία Εδάφους (*Course Over Ground*), σε δεκαδικές μοίρες
true_heading	Πορεία με βάση τον πραγματικό Βορρά, σε δεκαδικές μοίρες
nav_status	Κατάσταση πλοήγησης (κωδικός που υποδηλώνει την κατάσταση πλοήγησης του πλοίου: εν κινήσει, σταματημένο, κλπ.)
message_nr	Τύπος μηνύματος (κωδικός που υποδηλώνει το είδος του μηνύματος AIS: αναφορά θέσης, αναφορά σταθμού εδάφους, κλπ.)
dep_port	Λιμάνι αναχώρησης (αν είναι γνωστό)
arr_port	Λιμάνι προορισμού (αν είναι γνωστό)
curr_port	Τρέχον λιμάνι (αν το πλοίο βρίσκεται στην περιοχή ενός από τα λιμάνια της περιοχής μελέτης)

Πίνακας 5: Πεδία του κάθε AIS μηνύματος μετά τον εμπλουτισμό στην προ-επεξεργασία

Για την επεξεργασία των δεδομένων και την εκτέλεση των χωρικών ερωτημάτων, τα δεδομένα εισήχθησαν σε κατάλληλα διαμορφωμένη βάση δεδομένων PostgreSQL [12] με εγκατεστημένο το πρόσθετο χωρικής ανάλυσης PostGIS [13].

Για την προβολή των δεδομένων, την εξερεύνηση τους και την παράθεσή τους στο χάρτη, χρησιμοποιήθηκε το εργαλείο QGIS [14].

3.4 Υλοποίηση στο Spark Structured Streaming

Έχοντας προ-επεξεργαστεί το σύνολο δεδομένων μας, είμαστε έτοιμοι να το εισάγουμε στην εφαρμογή μας. Στον παρακάτω πίνακα (Πίνακας 6) παραθέτουμε τον ψευδοκώδικα που περιγράφει συνοπτικά τη ροή εργασίας μας εφαρμογής μας:

	Input: routes, waypoints, AIS messages
	Output: (time window start, time window end, vessel, trip, route, route score)
1	Set application parameters
2	Load: route names, waypoints
3	Create route segments
4	For each AIS message:
5	Replace null port values
6	Define trip ID
7	Fetch the routes (segments) for that trip
8	Calculate the distance of the vessel form each route segment
9	For each time window, vessel, trip ID:
10	Calculate the score for each route (median distance)

Πίνακας 6: Η ροή εργασίας της εφαρμογής μας

Αρχικά, ορίζουμε τις παραμέτρους της εφαρμογής, οι οποίες είναι:

- Η θέση του αρχείου *routes.csv* το οποίο περιέχει τα λεκτικά και τα αναγνωριστικά των 20 διαδρομών που χρησιμοποιούμε, μετά την προ-επεξεργασία (Πίνακας 1, Πίνακας 3).
- Η θέση του αρχείου *waypoints.csv*, το οποίο περιέχει τα σημεία των διαδρομών, μετά την προ-επεξεργασία (Πίνακας 4).
- Η θέση του καταλόγου *ais_data,* τον οποίο θα παρακολουθεί η εφαρμογή μας και όταν εισάγονται αρχεία με μηνύματα AIS (το αποτέλεσμα της προ-επεξεργασίας) στον κατάλογο αυτό, θα αρχίζει η παράλληλη επεξεργασία τους ως ροή δεδομένων (Πίνακας 5).
- Η θέση του καταλόγου εξόδου *out*, όπου θα αποθηκεύεται το τελικό αποτέλεσμα της επεξεργασίας (Πίνακας 7).
- Η θέση του καταλόγου *application_history*, τον οποίο η εφαρμογή μας θα χρησιμοποιεί ως checkpoint directory, για να αποθηκεύει το ιστορικό της επεξεργασίας ανά διαστήματα, κάτι που είναι χρήσιμο για την ανάκαμψη από σφάλματα.
- Τον αριθμό των partitions στα οποία θα χωρίσουμε το streaming DataFrame μας.
- Τη διάρκεια του (tumbling) window που εφαρμόζουμε στην επεξεργασία των δεδομένων μας. Το tumbling window είναι η χρονική περίοδος για την οποία θα γίνεται η συνάθροιση που υπολογίζει τη διάμεση απόσταση του κάθε πλοίου από την κάθε υποψήφια εναλλακτική διαδρομή του δρομολογίου που εκτελεί. Η διάρκεια που ορίζουμε αρχικά είναι 5 λεπτά.
- Τη διάρκεια του watermark που θα εφαρμόσουμε στην εφαρμογή μας. Το watermark είναι η ανώτατη χρονική διάρκεια αποδοχής των καθυστερημένων γεγονότων (μηνυμάτων) που εισέρχονται στην εφαρμογή μας. Το ορίζουμε στα 10 λεπτά. Αυτό σημαίνει ότι, κατά την επεξεργασία της ροής, μηνύματα παλαιότερα των 10 λεπτών (σύμφωνα με το πεδίο “date_time_utc” των μηνυμάτων), θα απορρίπτονται από το σύστημα κι έτσι θα βελτιστοποιούμε την χρήση των πόρων και θα ελευθερώνουμε τη μνήμη. Επειδή γενικά τα ταξίδια εντός της περιοχής μελέτης μας έχουν μικρή

διάρκεια, θεωρούμε αποδεκτό το χρονικό διάστημα 10 λεπτών διατήρησης των δεδομένων.

Η εφαρμογή μας ξεκινά με την δημιουργία του νέου Spark Session. Στη συνέχεια, ορίζουμε τα στατικά DataFrames των διαδρομών και των σημείων των διαδρομών.

Στην Εικόνα 14 φαίνεται το στατικό DataFrame που περιέχει τις διαδρομές. Το DataFrame το κρατάμε στη μνήμη με καλώντας τη μέθοδο cache().

route_id	route_type	route_name	dep_port	arr_port
1	1	EngeneNorth_In	null	Engene
2	1	EngeneNorth_Out	Engene	null
3	1	EngeneSouth_In	null	Engene
4	1	EngeneSouth_Out	Engene	null
5	1	MossNorth_In	null	Moss
6	1	MossNorth_Out	Moss	null
7	1	MossSouth_In	null	Moss
8	1	MossSouth_Out	Moss	null
9	1	OsloEast_In	null	Oslo
10	1	OsloEast_Out	Oslo	null
11	1	OsloWest_In	null	Oslo
12	1	OsloWest_Out	Oslo	null
13	1	HortenInner_In	null	Horten
14	1	HortenInner_Out	Horten	null
15	1	Horten_In	null	Horten
16	1	Horten_Out	Horten	null
17	2	Horten_Moss_North	Horten	Moss
18	2	Horten_Moss_South	Horten	Moss
19	2	Moss_Horten_North	Moss	Horten
20	2	Moss_Horten_South	Moss	Horten

Εικόνα 14: Το DataFrame με τις διαδρομές

Στην Εικόνα 15 φαίνεται το στατικό DataFrame που περιέχει τα waypoints των διαδρομών:

wp_id	route_id	rwp_id	wp_name	lon	lat
1	1	1	Hollenderbaen - c...	10.65851142	59.16198333
2	1	2	Gullholmen	10.56165578	59.43414435
3	1	3	Tofteholmen	10.59131467	59.51514138
4	1	4	Filtvet	10.63847067	59.56659672
5	1	5	Rammebaen	10.63629247	59.60851347
6	1	6	Kaholmen	10.61326418	59.6791572
7	1	7	Langebat	10.60634258	59.68498657
8	1	8	Storegrunnen	10.59217457	59.70973265
9	1	9	Haoya North	10.55261718	59.73194942
10	1	10	Nordre Sundbyholmen	10.53938332	59.72522943
11	1	11	Raudholmane	10.55521618	59.68393895
12	1	12	Engene	10.5449302	59.68388792
13	2	1	Engene	10.5449302	59.68388792
14	2	2	Raudholmane	10.55521618	59.68393895
15	2	3	Nordre Sundbyholmen	10.53938332	59.72522943
16	2	4	Haoya North	10.555401	59.7301014
17	2	5	Ristodden	10.57306255	59.71933663
18	2	6	Askholm	10.57536327	59.71214745
19	2	7	Tronstadodden	10.58558118	59.69885762
20	2	8	Kaholmen	10.61137912	59.67900408

only showing top 20 rows

Εικόνα 15: Το DataFrame με τα σημεία των διαδρομών

Κάθε waypoint έχει το δικό του γενικό αναγνωριστικό, αλλά και το αναγνωριστικό εντός της διαδρομής.

Στη συνέχεια, ορίζουμε το Schema, δηλαδή τα πεδία και τον τύπο τους, που θα έχει το streaming DataFrame το οποίο θα δημιουργηθεί από τα AIS μηνύματα. Στα streaming

DataFrames είναι υποχρεωτικό να παρέχουμε εκ τον προτέρων το Schema των δεδομένων, σε αντίθεση με τα στατικά DataFrames, όπου το schema μπορεί να εντοπιστεί και αυτόματα.

Στη συνέχεια, μετατρέπουμε τη στήλη “date_time_utc” (χρονική στιγμή) του κάθε μηνύματος από αλφαριθμητικό τύπο σε τύπου timestamp, ώστε να δουλέψει το tumbling window που θα ορίσουμε.

Στο σημείο αυτό, δεν γίνεται ακόμα καμία ενέργεια στο streaming DataFrame μας, παρά μόνο καταγράφονται τα transformations που θα εκτελεστούν σε αυτό. Οι υπολογισμοί θα αρχίσουν να γίνονται μόλις εκκινήσουμε το streaming query. Το τελικό αποτέλεσμα θα είναι η εγγραφή του τελικού αποτελέσματος στην έξοδο (sink) που έχουμε επιλέξει.

Ορίζουμε τα τρία DataFrames μας (routes, waypoints και messages) ως προσωρινά SQL Views, ώστε να μπορέσουμε να εκτελέσουμε ερωτήματα επεξεργασίας γραμμένα σε γλώσσα SQL.

Ακολουθεί η δημιουργία των ευθύγραμμων τμημάτων της κάθε διαδρομής. Αυτό γίνεται με το παρακάτω query (Εικόνα 16):

```
SELECT *
FROM   (SELECT wp_id,
               route_id,
               route_type,
               rwp_id,
               route_name,
               dep_port,
               arr_port,
               lon                  x1,
               lat                  y1,
               LEAD(lon, 1, 0)
                 OVER (
                   partition BY route_id
                   ORDER BY rwp_id) x2,
               LEAD(lat, 1, 0)
                 OVER (
                   partition BY route_id
                   ORDER BY rwp_id) y2
        FROM   (SELECT wp.wp_id,
                       wp.rwp_id,
                       rt.route_id,
                       rt.route_type,
                       rt.route_name,
                       rt.dep_port,
                       rt.arr_port,
                       wp.lon,
                       wp.lat
                FROM   waypoints wp
                       INNER JOIN routes rt
                               ON wp.route_id = rt.route_id) wp_raw) wp_lines
WHERE  x1 > 0
       AND y1 > 0
       AND x2 > 0
       AND y2 > 0
ORDER  BY 2,
          4
```

Εικόνα 16: Η δημιουργία των ευθύγραμμων τμημάτων

Εκτελούμε το Join (συγχώνευση) του waypoints DataFrame με το routes DataFrame. Καλούμε τη συνάρτηση LEAD() ώστε να συνδυάσουμε στην ίδια γραμμή το κάθε waypoint με το επόμενο του εντός της ίδιας διαδρομής, κι έτσι να ορίσουμε το ευθύγραμμο τμήμα.

Το αποτέλεσμα φαίνεται στην Εικόνα 17:

wp_id	route_id	route_type	rwp_id	route_name	dep_port	arr_port	x1	y1	x2	y2
1	1	1	1	EngeneNorth_In	null	Engene	10.65851142	59.16198333	10.56165578	59.43414435
2	1	1	2	EngeneNorth_In	null	Engene	10.56165578	59.43414435	10.59131467	59.51514138
3	1	1	3	EngeneNorth_In	null	Engene	10.59131467	59.51514138	10.63847067	59.56659672
4	1	1	4	EngeneNorth_In	null	Engene	10.63847067	59.56659672	10.63629247	59.60851347
5	1	1	5	EngeneNorth_In	null	Engene	10.63629247	59.60851347	10.61326418	59.6791572
6	1	1	6	EngeneNorth_In	null	Engene	10.61326418	59.6791572	10.60634258	59.68498657
7	1	1	7	EngeneNorth_In	null	Engene	10.60634258	59.68498657	10.59217457	59.70973265
8	1	1	8	EngeneNorth_In	null	Engene	10.59217457	59.70973265	10.55261718	59.73194942
9	1	1	9	EngeneNorth_In	null	Engene	10.55261718	59.73194942	10.53938332	59.72522943
10	1	1	10	EngeneNorth_In	null	Engene	10.53938332	59.72522943	10.55521618	59.68393895
11	1	1	11	EngeneNorth_In	null	Engene	10.55521618	59.68393895	10.5449302	59.68388792
13	2	1	1	EngeneNorth_Out	Engene	null	10.5449302	59.68388792	10.55521618	59.68393895
14	2	1	2	EngeneNorth_Out	Engene	null	10.55521618	59.68393895	10.53938332	59.72522943
15	2	1	3	EngeneNorth_Out	Engene	null	10.53938332	59.72522943	10.555401	59.7301014
16	2	1	4	EngeneNorth_Out	Engene	null	10.555401	59.7301014	10.57306255	59.71933663
17	2	1	5	EngeneNorth_Out	Engene	null	10.57306255	59.71933663	10.57536327	59.71214745
18	2	1	6	EngeneNorth_Out	Engene	null	10.57536327	59.71214745	10.58558118	59.69885762
19	2	1	7	EngeneNorth_Out	Engene	null	10.58558118	59.69885762	10.61137912	59.67900408
20	2	1	8	EngeneNorth_Out	Engene	null	10.61137912	59.67900408	10.61600285	59.66539057
21	2	1	9	EngeneNorth_Out	Engene	null	10.61600285	59.66539057	10.62806947	59.57108753

Εικόνα 17: Το DataFrame των ευθύγραμμων τμημάτων

Έχοντας ολοκληρώσει τους απαιτούμενους μετασχηματισμούς στο waypoints DataFrame, καλούμε τη μέθοδο broadcast(), ώστε να το διανείμουμε σε όλους τους κόμβους της συστάδας κι έτσι να υπάρχει διαθέσιμο στη μνήμη του κάθε κόμβου χωρίς να απαιτείται να υπάρξει ανταλλαγή εγγραφών μεταξύ των κόμβων (shuffling) κατά τη χρήση του. Υποθέτουμε ότι πάντοτε το waypoints DataFrame θα έχει μικρό αριθμό εγγραφών ώστε να διατηρείται στη μνήμη του κάθε κόμβου χωρίς να την επιβαρύνει.

Έπειτα, στο messages DataFrame, αντικαθιστούμε την τιμή null στα πεδία των λιμανιών, με την τιμή ”unknown” και δημιουργούμε τη στήλη “trip_id”, ως συνδυασμό των .τιμών των στηλών «λιμάνι_αναχώρησης_λιμάνι_προορισμού».

Στο σημείο αυτό, εκτελούμε το Join μεταξύ του streaming DataFrame messages και του static DataFrame waypoints, έτσι ώστε σε κάθε μήνυμα, να αντιστοιχίσουμε τα waypoints των διαδρομών που αφορούν τα συγκεκριμένα λιμάνια αναχώρησης και προορισμού. Αυτό γίνεται με το παρακάτω query (Εικόνα 18):

```
SELECT mmsi,
       imo_nr,
       v_length,
       tm,
       lon,
       lat,
       sog,
       cog,
       true_heading,
       nav_status,
       message_nr,
       msg.dep_port,
       msg.arr_port,
       curr_port,
       trip_id,
       trip_type,
       wp.wp_id,
       wp.route_id,
       wp.rwp_id,
       wp.route_name,
       wp.x1,
       wp.y1,
       wp.x2,
       wp.y2
FROM   messages msg
       INNER JOIN waypoints wp
                    ON ( msg.dep_port = wp.dep_port
                         OR msg.arr_port = wp.arr_port )
WHERE msg.dep_port <> msg.arr_port
```

Εικόνα 18: Αντιστοίχιση μηνυμάτων με διαδρομές

Στο σημείο αυτό, όσα μηνύματα εκπέμπονται από πλοία που βρίσκονται σε λιμάνι, θα απορριφθούν, διότι δεν υπάρχει διαδρομή στην οποία να είναι άγνωστο και το λιμάνι αναχώρησης, αλλά και το λιμάνι προορισμού. Θα απορριφθούν επίσης, μηνύματα που εκπέμπονται από πλοία που αναχωρούν από ένα λιμάνι κι επιστρέφουν στο ίδιο.

Έτσι, είμαστε πλέον έτοιμοι να υπολογίσουμε την απόσταση της κάθε θέσης πλοίου, με το κάθε σημείο της υποψήφιας διαδρομής. Πριν το κάνουμε αυτό, κάνουμε repartition στο DataFrame messages, με βάση τη στήλη mmsi. Έτσι, όλες οι εγγραφές που αφορούν το ίδιο πλοίο θα καταλήξουν στον ίδιο κόμβο του cluster, κι έτσι, ο υπολογισμός της απόστασης μεταξύ των θέσεων των πλοίων και των ευθυγράμμων τμημάτων των διαδρομών, θα πραγματοποιηθεί ανεξάρτητα στον κάθε κόμβο.

Αφού υπολογίσουμε την απόσταση σύμφωνα με τη μετρική που έχουμε επιλέξει, έχουμε ένα νέο DataFrame με την επιπλέον στήλη της απόστασης.

Είμαστε πλέον σε θέση να εκτελέσουμε το τελικό query επεξεργασίας, στο οποίο, ανά διαστήματα, θα υπολογίζουμε τη διάμεση απόσταση του κάθε πλοίου από την κάθε διαδρομή. Το query αυτό φαίνεται στην Εικόνα 19:

```
SELECT window.start,
       window.end,
       mmsi,
       trip_id,
       route_name,
       Percentile_approx(distance, 0.5) AS med_dist
FROM   messages
WHERE  trip_id <> 'unknown_unknown'
GROUP  BY Window(tm, '""" + window_duration + """'),
          mmsi,
          trip_id,
          route_name
```

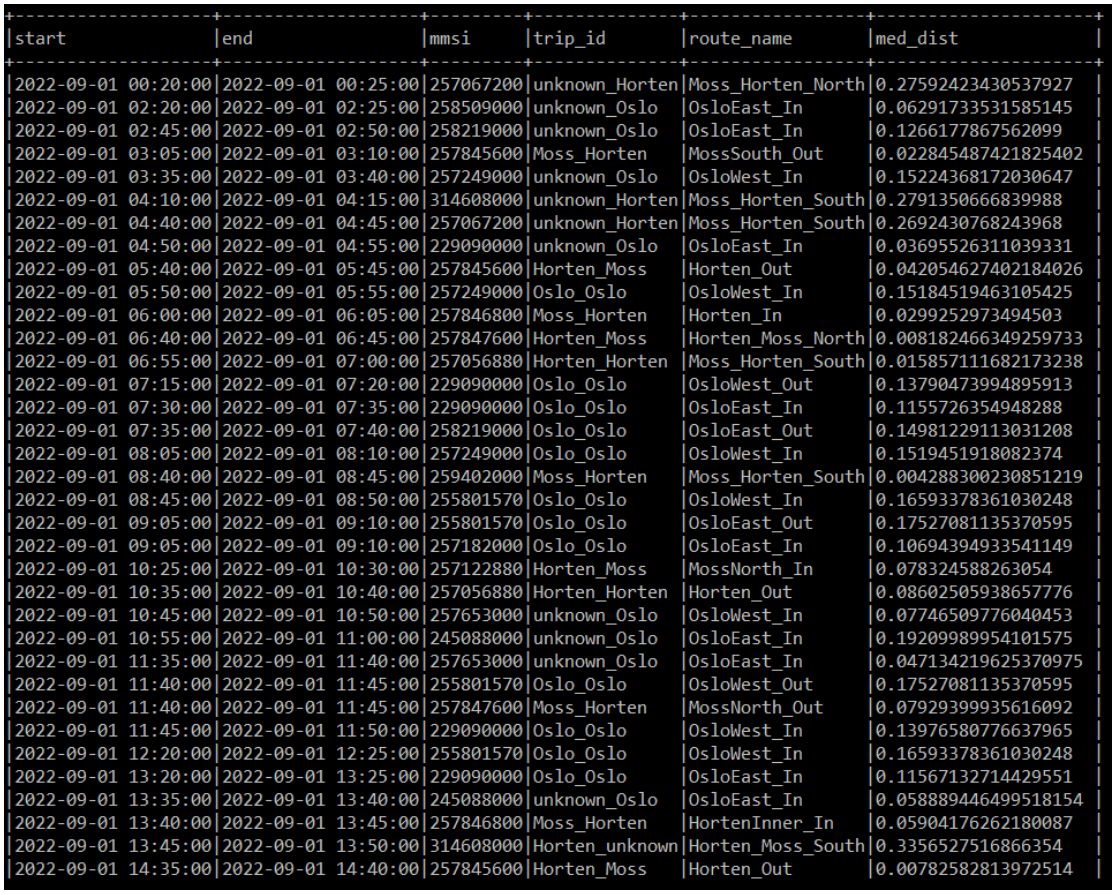
Εικόνα 19: Βαθμολόγηση δρομολογίου ανά χρονικό διάστημα

Κάνουμε group by ανά χρονικό παράθυρο που έχουμε επιλέξει, ανά mmsi, trip_id και route_name.

Τέλος, ορίζουμε το streaming query και τις απαραίτητες παραμέτρους εισόδου και εξόδου. Ορίζουμε το append ως τρόπο λειτουργίας εξόδου, το οποίο, σε συνδυασμό με το watermark που εφαρμόζουμε, θα εμφανίζει μόνο τις νέες εγγραφές αποτελεσμάτων μετά την επεξεργασία του κάθε batch.

Η επεξεργασία της ροής δεδομένων ξεκινά μόλις η εφαρμογή μας εντοπίσει αρχεία μηνυμάτων στον κατάλογο εισόδου *ais_data*.

Στην επόμενη εικόνα (Εικόνα 20) προβάλουμε ένα τμήμα των αποτελεσμάτων έχοντας επιλέξει ως έξοδο (sink) την κονσόλα του συστήματος:



start	end	mmsi	trip_id	route_name	med_dist
2022-09-01 00:20:00	2022-09-01 00:25:00	257067200	unknown_Horten	Moss_Horten_North	0.27592423430537927
2022-09-01 02:20:00	2022-09-01 02:25:00	258509000	unknown_Oslo	OsloEast_In	0.06291733531585145
2022-09-01 02:45:00	2022-09-01 02:50:00	258219000	unknown_Oslo	OsloEast_In	0.1266177867562099
2022-09-01 03:05:00	2022-09-01 03:10:00	257845600	Moss_Horten	MossSouth_Out	0.022845487421825402
2022-09-01 03:35:00	2022-09-01 03:40:00	257249000	unknown_Oslo	OsloWest_In	0.15224368172030647
2022-09-01 04:10:00	2022-09-01 04:15:00	314608000	unknown_Horten	Moss_Horten_South	0.2791350666839988
2022-09-01 04:40:00	2022-09-01 04:45:00	257067200	unknown_Horten	Moss_Horten_South	0.2692430768243968
2022-09-01 04:50:00	2022-09-01 04:55:00	229090000	unknown_Oslo	OsloEast_In	0.03695526311039331
2022-09-01 05:40:00	2022-09-01 05:45:00	257845600	Horten_Moss	Horten_Out	0.042054627402184026
2022-09-01 05:50:00	2022-09-01 05:55:00	257249000	Oslo_Oslo	OsloWest_In	0.15184519463105425
2022-09-01 06:00:00	2022-09-01 06:05:00	257846800	Moss_Horten	Horten_In	0.0299252973494503
2022-09-01 06:40:00	2022-09-01 06:45:00	257847600	Horten_Moss	Horten_Moss_North	0.008182466349259733
2022-09-01 06:55:00	2022-09-01 07:00:00	257056880	Horten_Horten	Moss_Horten_South	0.015857111682173238
2022-09-01 07:15:00	2022-09-01 07:20:00	229090000	Oslo_Oslo	OsloWest_Out	0.13790473994895913
2022-09-01 07:30:00	2022-09-01 07:35:00	229090000	Oslo_Oslo	OsloEast_In	0.1155726354948288
2022-09-01 07:35:00	2022-09-01 07:40:00	258219000	Oslo_Oslo	OsloEast_Out	0.14981229113031208
2022-09-01 08:05:00	2022-09-01 08:10:00	257249000	Oslo_Oslo	OsloWest_In	0.1519451918082374
2022-09-01 08:40:00	2022-09-01 08:45:00	259402000	Moss_Horten	Moss_Horten_South	0.004288300230851219
2022-09-01 08:45:00	2022-09-01 08:50:00	255801570	Oslo_Oslo	OsloWest_In	0.16593378361030248
2022-09-01 09:05:00	2022-09-01 09:10:00	255801570	Oslo_Oslo	OsloEast_Out	0.17527081135370595
2022-09-01 09:05:00	2022-09-01 09:10:00	257182000	Oslo_Oslo	OsloEast_In	0.10694394933541149
2022-09-01 10:25:00	2022-09-01 10:30:00	257122880	Horten_Moss	MossNorth_In	0.078324588263054
2022-09-01 10:35:00	2022-09-01 10:40:00	257056880	Horten_Horten	Horten_Out	0.08602505938657776
2022-09-01 10:45:00	2022-09-01 10:50:00	257653000	unknown_Oslo	OsloWest_In	0.07746509776040453
2022-09-01 10:55:00	2022-09-01 11:00:00	245088000	unknown_Oslo	OsloEast_In	0.19209989954101575
2022-09-01 11:35:00	2022-09-01 11:40:00	257653000	unknown_Oslo	OsloEast_In	0.047134219625370975
2022-09-01 11:40:00	2022-09-01 11:45:00	255801570	Oslo_Oslo	OsloWest_Out	0.17527081135370595
2022-09-01 11:40:00	2022-09-01 11:45:00	257847600	Moss_Horten	MossNorth_Out	0.07929399935616092
2022-09-01 11:45:00	2022-09-01 11:50:00	229090000	Oslo_Oslo	OsloWest_In	0.13976580776637965
2022-09-01 12:20:00	2022-09-01 12:25:00	255801570	Oslo_Oslo	OsloWest_In	0.16593378361030248
2022-09-01 13:20:00	2022-09-01 13:25:00	229090000	Oslo_Oslo	OsloEast_In	0.11567132714429551
2022-09-01 13:35:00	2022-09-01 13:40:00	245088000	unknown_Oslo	OsloEast_In	0.058889446499518154
2022-09-01 13:40:00	2022-09-01 13:45:00	257846800	Moss_Horten	HortenInner_In	0.05904176262180087
2022-09-01 13:45:00	2022-09-01 13:50:00	314608000	Horten_unknown	Horten_Moss_South	0.3356527516866354
2022-09-01 14:35:00	2022-09-01 14:40:00	257845600	Horten_Moss	Horten_Out	0.00782582813972514

Εικόνα 20: Το DataFrame των αποτελεσμάτων της streaming εφαρμογής

Κάθε γραμμή του αποτελέσματος περιέχει την αρχή και το τέλος του χρονικού διαστήματος της ανάλυσης, το μοναδικό αναγνωριστικό του πλοίου, το τρέχον ταξίδι, την υποψήφια εναλλακτική διαδρομή και τη διάμεση απόσταση του πλοίου από αυτήν (Πίνακας 7).

Όνομα Πεδίου	Χρήση
start	Η αρχή του τρέχοντος χρονικού διαστήματος ανάλυσης
end	Το τέλος του τρέχοντος χρονικού διαστήματος ανάλυσης
mmsi	Αναγνωριστικό του πλοίου για σκοπούς επικοινωνίας
trip_id	Αναγνωριστικό του τρέχοντος ταξιδιού
route_name	Το όνομα της διαδρομής
med_dist	Η διάμεσος των αποστάσεων των θέσεων του πλοίου από τα ευθύγραμμα τμήματα της διαδρομής, για το τρέχον χρονικό διάστημα και ταξίδι. Η διαδρομή με τη χαμηλότερη τιμή στο πεδίο αυτό, είναι περισσότερο πιθανό να ακολουθούνται στο τρέχον ταξίδι.

Πίνακας 7: Πεδία του πίνακα/αρχείου εξόδου της streaming εφαρμογής

Ιδανικά, θα επιθυμούσαμε τα αποτελέσματα να είναι ταξινομημένα ως προς το χρονικό διάστημα, το mmsi, και το σκορ της κάθε διαδρομής, όμως η ταξινόμηση δεν είναι δυνατό να γίνει σε streaming DataFrame, με χρήση watermark και append output mode. Για να γίνει αυτό, θα έπρεπε να είχαμε ορίσει complete output mode, η οποία όμως διατηρεί όλα τα προηγούμενα δεδομένα στη μνήμη (αγνοώντας το watermark, αν το έχουμε ορίσει). Το complete output mode θα επιβάρυνε τελικά τους πόρους των κόμβων, διατηρώντας στη μνήμη ένα απροσδιόριστο αριθμό εγγραφών που συνεχώς αυξάνονται.

4 Πειραματική Μελέτη

Στο κεφάλαιο αυτό, πραγματοποιούμε εκτέλεση της εφαρμογής ορίζοντας διάφορες παραμέτρους, ώστε να αξιολογήσομε την απόδοση του συστήματος.

Οι δοκιμές εκτελέστηκαν σε ένα τοπικό μηχάνημα με εγκατεστημένο το Apache Spark έκδοσης 3.1.1 και τα ακόλουθα χαρακτηριστικά:

- Λειτουργικό Σύστημα: Microsoft Windows 10
- CPU: 4 φυσικοί πυρήνες (8 λογικοί πυρήνες) συχνότητας 3.4 GHz
- RAM: 32 GB

Το βασικό σύνολο δεδομένων περιέχει 1.4 εκατομμύρια εγγραφές, και από αυτό, δημιουργήσαμε τρία υποσύνολα. Στον παρακάτω πίνακα (Πίνακας 8) φαίνονται τα χαρακτηριστικά των υποσυνόλων δεδομένων που χρησιμοποιήσαμε:

Dataset	Αριθμός εγγραφών	Ποσοστό % του συνόλου
A	355165	25%
B	710329	50%
C	1065515	75%
D	1420687	100%

Πίνακας 8: Μεγέθη συνόλων δεδομένων πειραμάτων

Εκτελέσαμε την εφαρμογή μας με είσοδο καθένα από τα παραπάνω σύνολα δεδομένων και κάναμε δοκιμές θέτοντας, για κάθε σύνολο, τους εξής συνδυασμούς παραμέτρων (Πίνακας 9 και Πίνακας 10):

Παράμετρος	Τιμές			
Αριθμός partitions	4	8	16	150

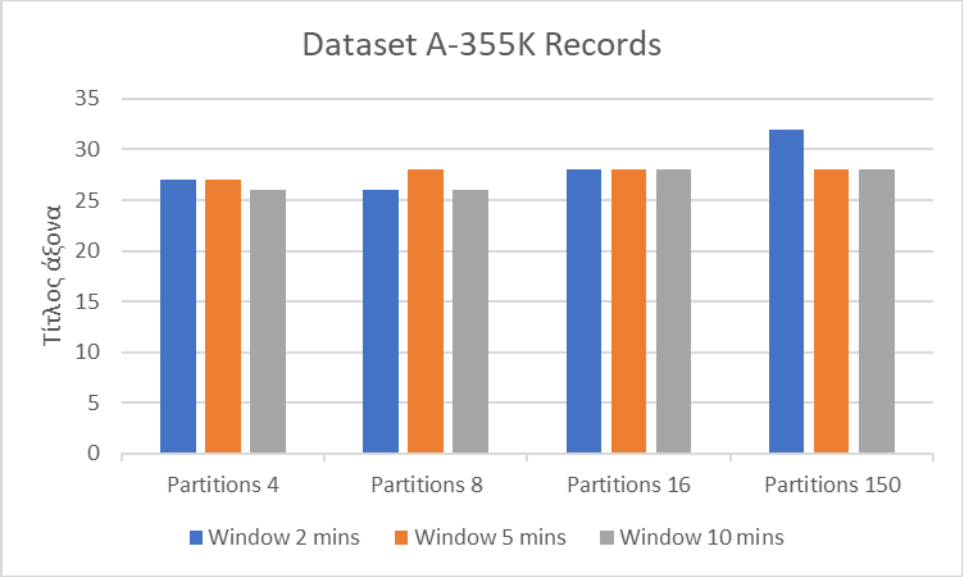
Πίνακας 9: Τιμές παραμέτρου αριθμού partitions

Παράμετρος	Τιμές		
Διάρκεια Window	2 λεπτά	5 λεπτά	10 λεπτά

Πίνακας 10: Τιμές διάρκειας χρονικού παραθύρου

Η βασική μετρική αποτίμησης της απόδοσης, είναι ο χρόνος επεξεργασίας σε δευτερόλεπτα. Δώσαμε κάθε υποσύνολο ως είσοδο στην εφαρμογή μας, σε μορφή αρχείου CSV, και η εφαρμογή το επεξεργάστηκε ως ένα batch. Οι χρόνοι εκτέλεσης που παρουσιάζονται παρακάτω, αφορούν το χρόνο επεξεργασίας αυτού του batch δεδομένων (streaming DataFrame) όπως αυτός αναφέρεται από τα εργαλεία παρακολούθησης που παρέχει Spark, και δεν περιλαμβάνουν το χρόνο εγγραφής των αποτελεσμάτων στην έξοδο.

Στην επόμενη εικόνα (Εικόνα 21) και πίνακα (Πίνακας 11) φαίνονται οι χρόνοι επεξεργασίας για το Dataset A, ανά αριθμό partitions και διάρκεια του χρονικού παραθύρου:

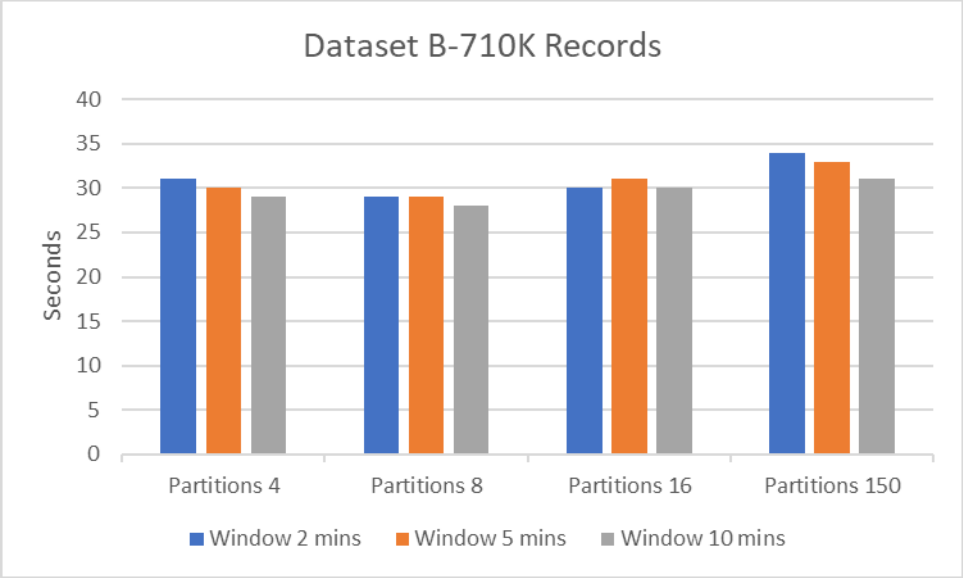


Εικόνα 21: Χρόνος επεξεργασίας υποσυνόλου δεδομένων Α

Dataset A	Window 2 mins	Window 5 mins	Window 10 mins
Partitions 4	27	27	26
Partitions 8	26	28	26
Partitions 16	28	28	28
Partitions 150	32	28	28

Πίνακας 11: Χρόνος επεξεργασίας υποσυνόλου δεδομένων Α

Στην επόμενη εικόνα (Εικόνα 22) και πίνακα (Πίνακας 12) φαίνονται οι χρόνοι επεξεργασίας για το Dataset B, ανά αριθμό partitions και διάρκεια του χρονικού παραθύρου:

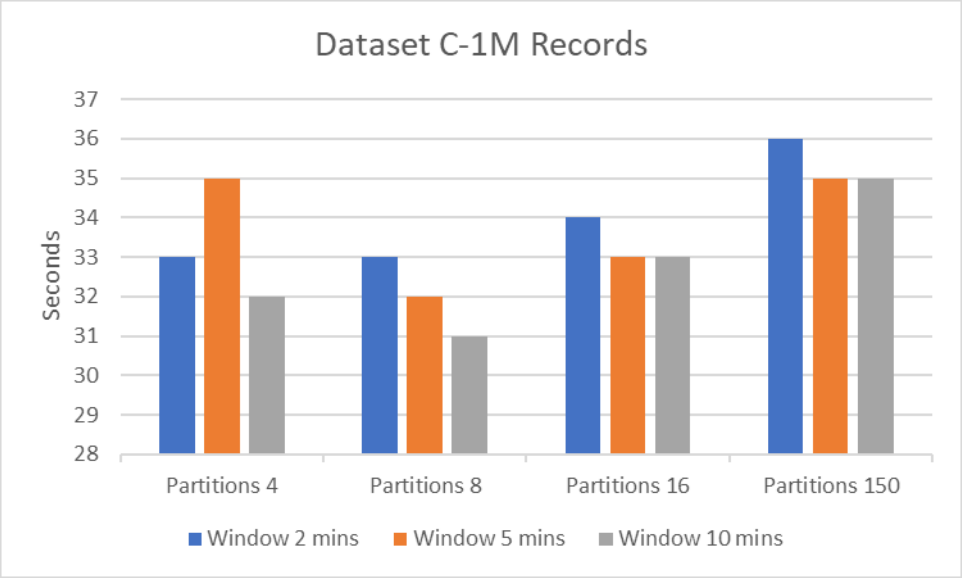


Εικόνα 22: Χρόνος επεξεργασίας υποσυνόλου δεδομένων Β

Dataset B	Window 2 mins	Window 5 mins	Window 10 mins
Partitions 4	31	30	29
Partitions 8	29	29	28
Partitions 16	30	31	30
Partitions 150	34	33	31

Πίνακας 12: Χρόνος επεξεργασίας υποσυνόλου δεδομένων B

Στην επόμενη εικόνα (Εικόνα 23) και πίνακα (Πίνακας 13) φαίνονται οι χρόνοι επεξεργασίας για το Dataset C, ανά αριθμό partitions και διάρκεια του χρονικού παραθύρου:

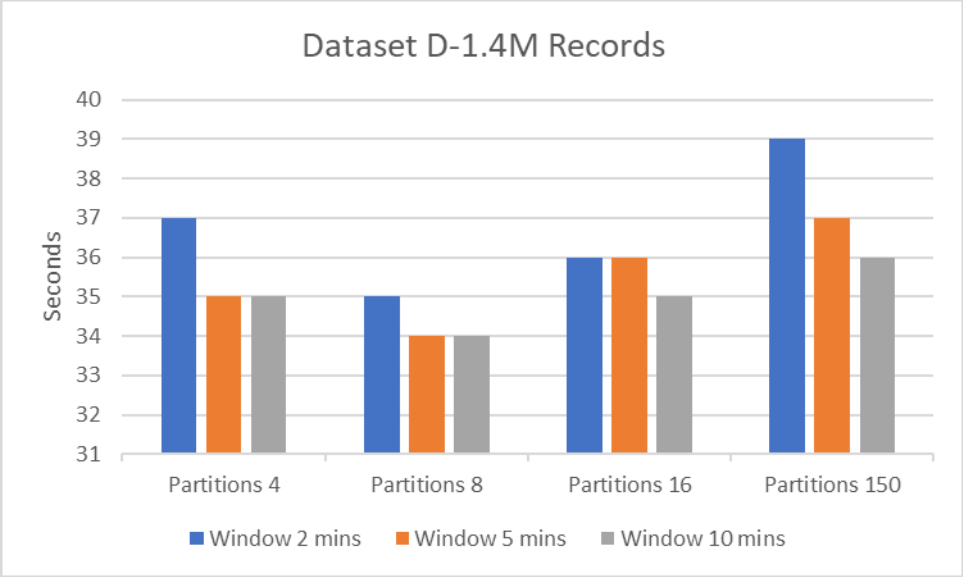


Εικόνα 23: Χρόνος επεξεργασίας υποσυνόλου δεδομένων C

Dataset C	Window 2 mins	Window 5 mins	Window 10 mins
Partitions 4	33	35	32
Partitions 8	33	32	31
Partitions 16	34	33	33
Partitions 150	36	35	35

Πίνακας 13: Χρόνος επεξεργασίας υποσυνόλου δεδομένων C

Στην επόμενη εικόνα (Εικόνα 24) και πίνακα (Πίνακας 14) φαίνονται οι χρόνοι επεξεργασίας για το Dataset D, ανά αριθμό partitions και διάρκεια του χρονικού παραθύρου:



Εικόνα 24: Χρόνος επεξεργασίας υποσυνόλου δεδομένων D

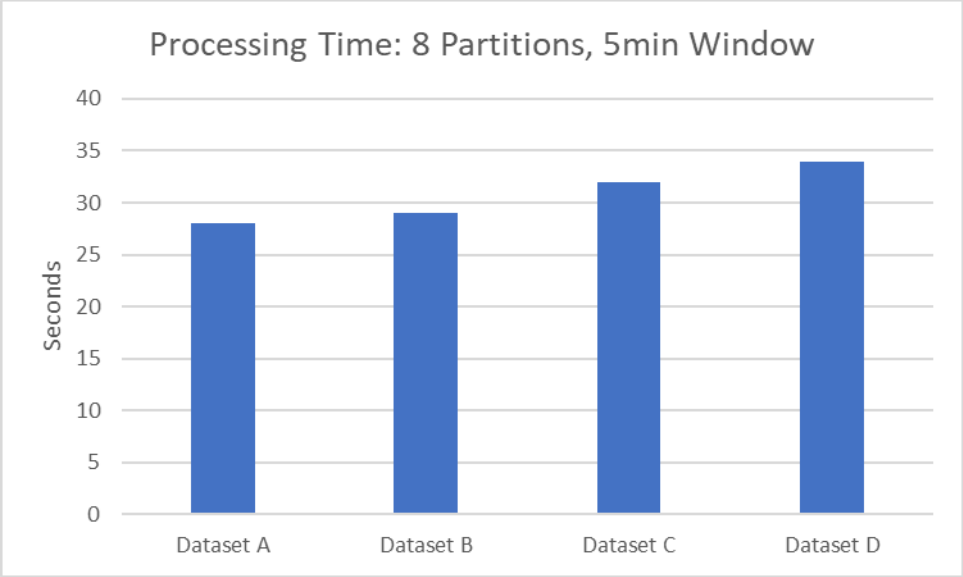
Dataset D	Window 2 mins	Window 5 mins	Window 10 mins
Partitions 4	37	35	35
Partitions 8	35	34	34
Partitions 16	36	36	35
Partitions 150	39	37	36

Πίνακας 14: Χρόνος επεξεργασίας υποσυνόλου δεδομένων D

Παρατηρούμε ότι γενικά, πετυχαίνουμε μικρότερο χρόνο επεξεργασίας όταν ορίζουμε 8 partitions, όσα δηλαδή και οι λογικοί επεξεργαστές του συστήματός μας. Λόγω του ότι αξιολογούνται όλοι οι πυρήνες του συστήματός μας, επιτυγχάνεται μεγαλύτερη παραλληλία κι έτσι μειώνεται ο χρόνος επεξεργασίας. Όταν έχουμε λιγότερα partitions (4) δεν αξιοποιούνται όλοι οι διαθέσιμοι λογικοί πυρήνες, ενώ σε περισσότερα partitions, ο χρόνος επεξεργασίας αυξάνεται λόγω της επιπλέον επιβάρυνσης και του κόστους από την εναλλαγή των πυρήνων κατά την ανάληψη των tasks από τους workers.

Παρατηρούμε επίσης ότι ο σημαντικότερος παράγοντας που επηρεάζει το χρόνο επεξεργασίας είναι το μέγεθος του συνόλου δεδομένων, ενώ οι διαφορές στο window duration δεν επιφέρουν τελικά μεγάλες διαφορές στο χρόνο επεξεργασίας.

Στην επόμενη εικόνα (Εικόνα 25) και πίνακα (Πίνακας 15) εστιάζουμε στο χρόνο επεξεργασίας για 8 partitions και διάρκεια χρονικού παραθύρου 5 λεπτά.



Εικόνα 25: Σύγκριση χρόνου επεξεργασίας υποσυνόλων δεδομένων

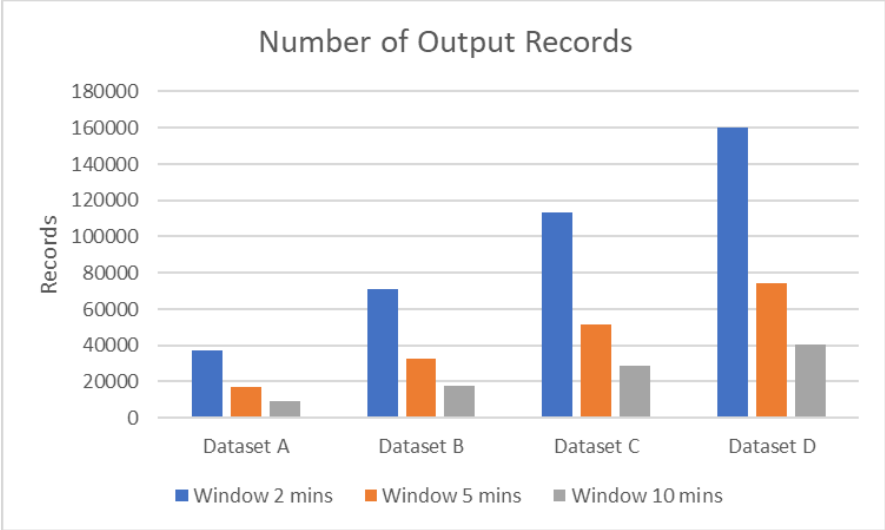
Dataset	Χρόνος επεξεργασίας (seconds)
A	28
B	29
C	32
D	34

Πίνακας 15: Σύγκριση χρόνου επεξεργασίας υποσυνόλων δεδομένων

Ο χρόνος επεξεργασίας αυξάνεται όταν επεξεργαζόμαστε μεγαλύτερο σύνολο δεδομένων, όμως δεν παρατηρούμε μεγάλες διαφορές. Ο χρόνος επεξεργασίας για το Dataset D (34 δευτερόλεπτα), είναι κατά 21.42% μεγαλύτερος σε σχέση με το χρόνο επεξεργασίας για το Dataset A (28 δευτερόλεπτα).

Έχουμε υπόψιν μας ότι υπάρχουν μηνύματα AIS που αφορούν πλοία που βρίσκονται σε λιμάνι ή με άγνωστο λιμάνι αναχώρησης και προορισμού, με αποτέλεσμα αυτά τα μηνύματα να απορριφθούν σχετικά νωρίς κατά την επεξεργασία.

Τέλος, στην επόμενη εικόνα (Εικόνα 26) και πίνακα (Πίνακας 16) παραθέτουμε τον αριθμό των εγγραφών που παράγονται στην έξοδο για κάθε σύνολο δεδομένων.



Εικόνα 26: Σύγκριση αριθμού αποτελεσμάτων

Dataset	Window 2 mins	Window 5 mins	Window 10 mins
A	37002	16936	9362
B	71188	32558	17994
C	113408	51830	28538
D	159866	74156	40692

Πίνακας 16: Σύγκριση αριθμού αποτελεσμάτων

Είναι ξεκάθαρο ότι όσο αυξάνεται η διάρκεια του χρονικού παραθύρου, δημιουργούνται λιγότερες εγγραφές στην έξοδο, αφού η κάθε συνάθροιση λαμβάνει υπόψη περισσότερες εγγραφές και η χρονική περίοδος των 10 ημερών χωρίζεται σε μεγαλύτερα χρονικά διαστήματα.

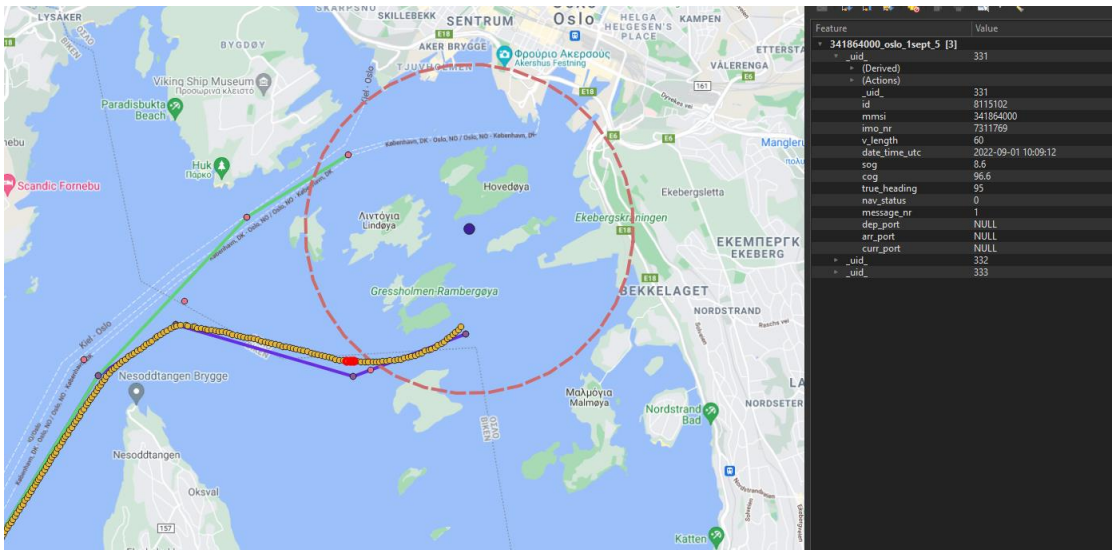
5 Μελέτες Περιπτώσεων

Στο κεφάλαιο αυτό, ασχολούμαστε με την παρουσίαση κάποιων περιπτώσεων στα δεδομένα και τα αντίστοιχα αποτελέσματα του αλγόριθμου. Παραθέτουμε χάρτες όπου απεικονίζονται οι τροχιές έτσι όπως καταγράφονται από τα μηνύματα AIS, και βλέπουμε ποιο υποψήφιο δρομολόγιο έχει το καλύτερο σκορ (δηλαδή το πλοίο απέχει τη μικρότερη διάμεση απόσταση). Στον Πίνακας 1 περιγράφονται οι χρωματικοί κωδικοί των διαδρομών.

Για τη μελέτη αυτή των αποτελεσμάτων και την επιβεβαίωση της ορθής λειτουργίας του αλγόριθμου, χρησιμοποιήσαμε το Dataset D, δηλαδή όλες τις διαθέσιμες εγγραφές, και το επεξεργαστήκαμε σε λειτουργία batch και όχι streaming. Αυτό, κρίθηκε απαραίτητο διότι επιθυμούμε κάθε φορά να δούμε άμεσα ποιο είναι το «καλύτερο» εναλλακτικό δρομολόγιο ανά πλοίο, ταξίδι και χρονική περίοδο, κι έτσι πρέπει να εφαρμόσουμε ταξινόμηση (sorting) στα δεδομένα, και στη συνέχεια να κάνουμε άλλη μια συνάθροιση για να προβάλουμε μόνο ένα δρομολόγιο από τα εναλλακτικά. Το streaming DataFrame όμως δεν επιτρέπει πολλαπλές συναθροίσεις έπειτα από μία συνάθροιση στο ίδιο DataFrame, κι έτσι δημιουργήσαμε μια batch εφαρμογή για να μπορέσουμε να εκτελέσουμε την επιπλέον συνάθροιση.

5.1 Διαδρομή OsloEast_In

Το πλοίο με mmsi 341864000 κατά το ταξίδι του από τις 2022-09-01 09:45:00 έως τις 2022-09-01 10:15:00, εντοπίζεται να ακολουθεί τη διαδρομή OsloEast_In (μοβ διαδρομή), όπως φαίνεται στην Εικόνα 27 και στην Εικόνα 28:



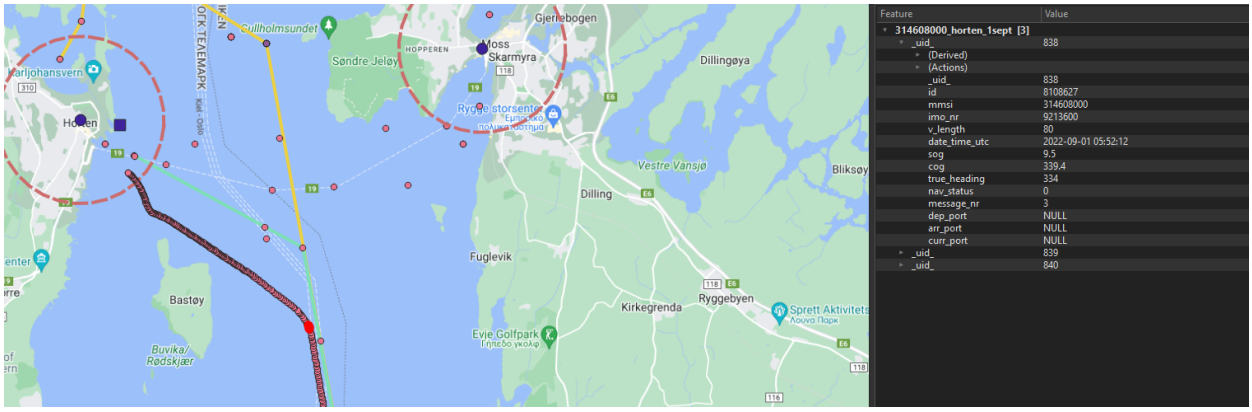
Εικόνα 27: Διαδρομή OsloEast_In

start	end	mmsi	trip_id	route_name
2022-09-01 09:45:00	2022-09-01 09:50:00	341864000	unknown_Oslo	OsloEast_In
2022-09-01 09:50:00	2022-09-01 09:55:00	341864000	unknown_Oslo	OsloEast_In
2022-09-01 09:55:00	2022-09-01 10:00:00	341864000	unknown_Oslo	OsloEast_In
2022-09-01 10:00:00	2022-09-01 10:05:00	341864000	unknown_Oslo	OsloEast_In
2022-09-01 10:05:00	2022-09-01 10:10:00	341864000	unknown_Oslo	OsloEast_In
2022-09-01 10:10:00	2022-09-01 10:15:00	341864000	unknown_Oslo	OsloEast_In

Εικόνα 28: Διαδρομή OsloEast_In - αποτελέσματα

5.2 Διαδρομή Horten_In

Το πλοίο με mmsi 314608000 κατά το ταξίδι του από τις 2022-09-01 05:00:00 έως τις 2022-09-01 06:20:00, εντοπίζεται να ακολουθεί τη διαδρομή Horten_In (πράσινη διαδρομή), όπως φαίνεται στην Εικόνα 29 και στην Εικόνα 30:



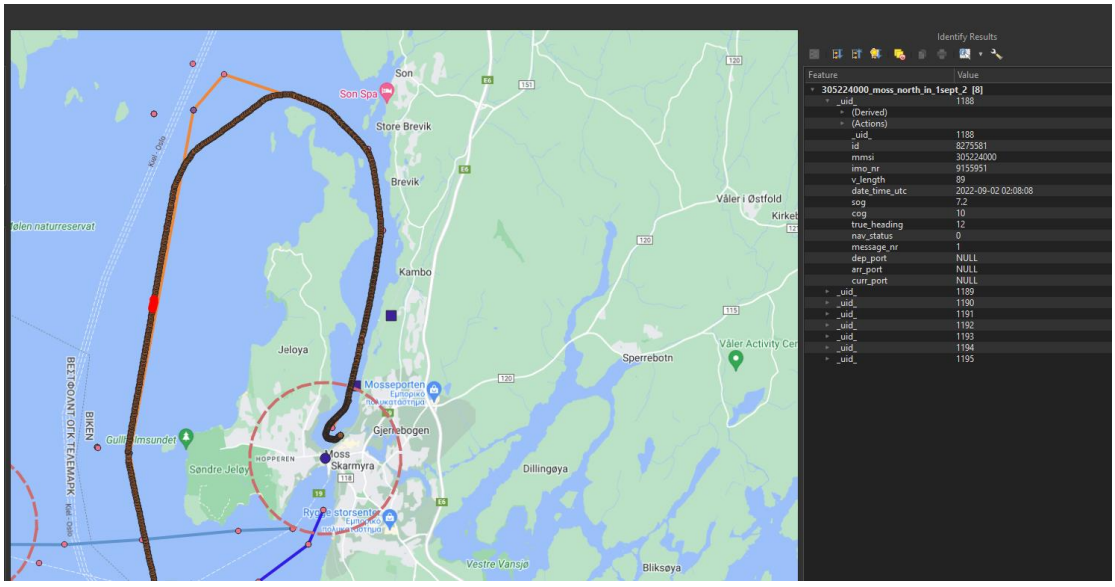
Εικόνα 29: Διαδρομή Horten_In

start	end	mmsi	trip_id	route_name
2022-09-01 05:05:00	2022-09-01 05:10:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:10:00	2022-09-01 05:15:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:15:00	2022-09-01 05:20:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:20:00	2022-09-01 05:25:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:25:00	2022-09-01 05:30:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:30:00	2022-09-01 05:35:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:35:00	2022-09-01 05:40:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:40:00	2022-09-01 05:45:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:45:00	2022-09-01 05:50:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:50:00	2022-09-01 05:55:00	314608000	unknown_Horten	Horten_In
2022-09-01 05:55:00	2022-09-01 06:00:00	314608000	unknown_Horten	Horten_In
2022-09-01 06:00:00	2022-09-01 06:05:00	314608000	unknown_Horten	Horten_In
2022-09-01 06:05:00	2022-09-01 06:10:00	314608000	unknown_Horten	Horten_In
2022-09-01 06:10:00	2022-09-01 06:15:00	314608000	unknown_Horten	Horten_In
2022-09-01 06:15:00	2022-09-01 06:20:00	314608000	unknown_Horten	Horten_In

Εικόνα 30: Διαδρομή Horten_In - αποτελέσματα

5.3 Διαδρομή MossNorth_In

Το πλοίο με mmsi 305224000 κατά το ταξίδι του από τις 2022-09-02 02:10:00 έως τις 2022-09-02 02:55:00, εντοπίζεται να ακολουθεί τη διαδρομή MossNorth_In (πορτοκαλί διαδρομή), όπως φαίνεται στην Εικόνα 31 και στην Εικόνα 32:



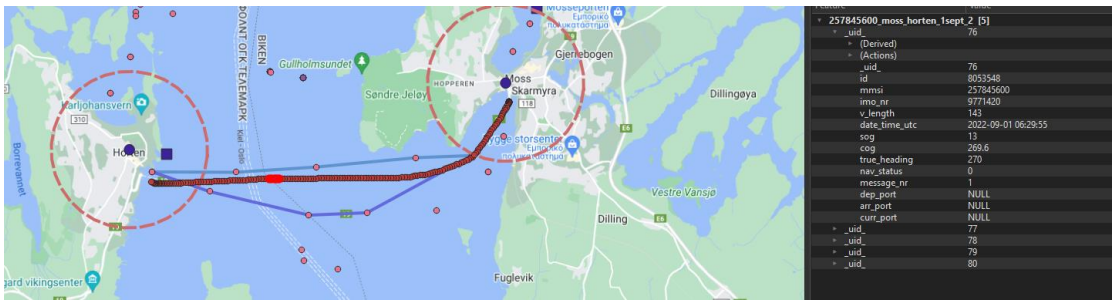
Εικόνα 31: Διαδρομή MossNorth_In

start	end	mmsi	trip_id	route_name
2022-09-02 01:55:00	2022-09-02 02:00:00	305224000	unknown_Moss	Horten_Moss_North
2022-09-02 02:00:00	2022-09-02 02:05:00	305224000	unknown_Moss	Horten_Moss_North
2022-09-02 02:05:00	2022-09-02 02:10:00	305224000	unknown_Moss	Horten_Moss_North
2022-09-02 02:10:00	2022-09-02 02:15:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:15:00	2022-09-02 02:20:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:20:00	2022-09-02 02:25:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:25:00	2022-09-02 02:30:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:30:00	2022-09-02 02:35:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:35:00	2022-09-02 02:40:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:40:00	2022-09-02 02:45:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:45:00	2022-09-02 02:50:00	305224000	unknown_Moss	MossNorth_In
2022-09-02 02:50:00	2022-09-02 02:55:00	305224000	unknown_Moss	MossNorth_In

Εικόνα 32: Διαδρομή MossNorth_In - αποτελέσματα

5.4 Διαδρομή Moss_Horten_North

Το πλοίο με mmsi 257845600 κατά το ταξίδι του από τις 2022-09-01 06:15:00 έως τις 2022-09-01 06:35:00, εντοπίζεται να ακολουθεί τη διαδρομή Moss_Horten_North (γαλάζια διαδρομή), όπως φαίνεται στην Εικόνα 33 και στην Εικόνα 34:



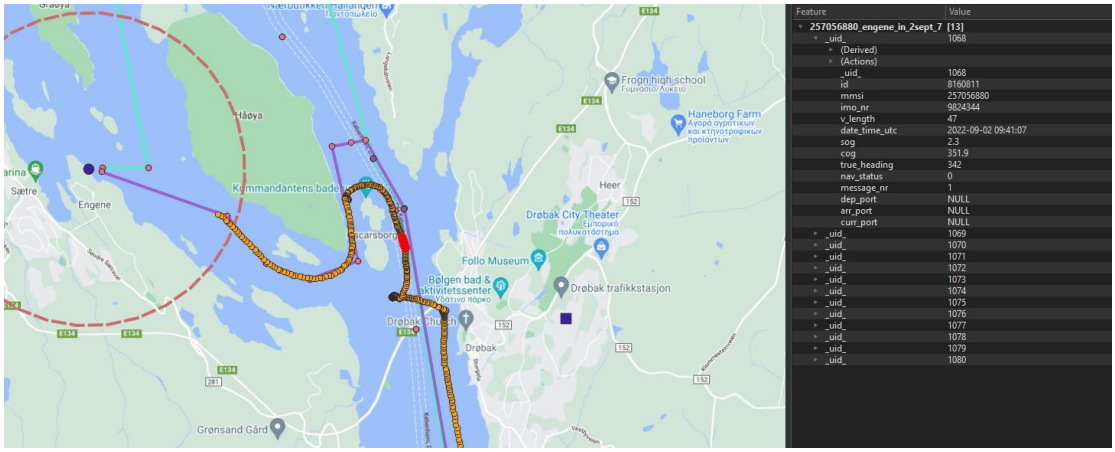
Εικόνα 33: διαδρομή Moss_Horten_North

start	end	mmsi	trip_id	route_name
2022-09-01 06:15:00	2022-09-01 06:20:00	257845600	Moss_Horten	Moss_Horten_North
2022-09-01 06:20:00	2022-09-01 06:25:00	257845600	Moss_Horten	Moss_Horten_North
2022-09-01 06:25:00	2022-09-01 06:30:00	257845600	Moss_Horten	Moss_Horten_North
2022-09-01 06:30:00	2022-09-01 06:35:00	257845600	Moss_Horten	Moss_Horten_North

Εικόνα 34: Διαδρομή Moss_Horten_North - αποτελέσματα

5.5 Διαδρομή EngeneSouth_In

Το πλοίο με mmsi 257056880 κατά το ταξίδι του από τις 2022-09-02 09:05:00 έως τις 2022-09-02 12:00:00, εντοπίζεται να ακολουθεί τη διαδρομή EngeneSouth_In (μοβ διαδρομή), όπως φαίνεται στην Εικόνα 35 και στην Εικόνα 36:



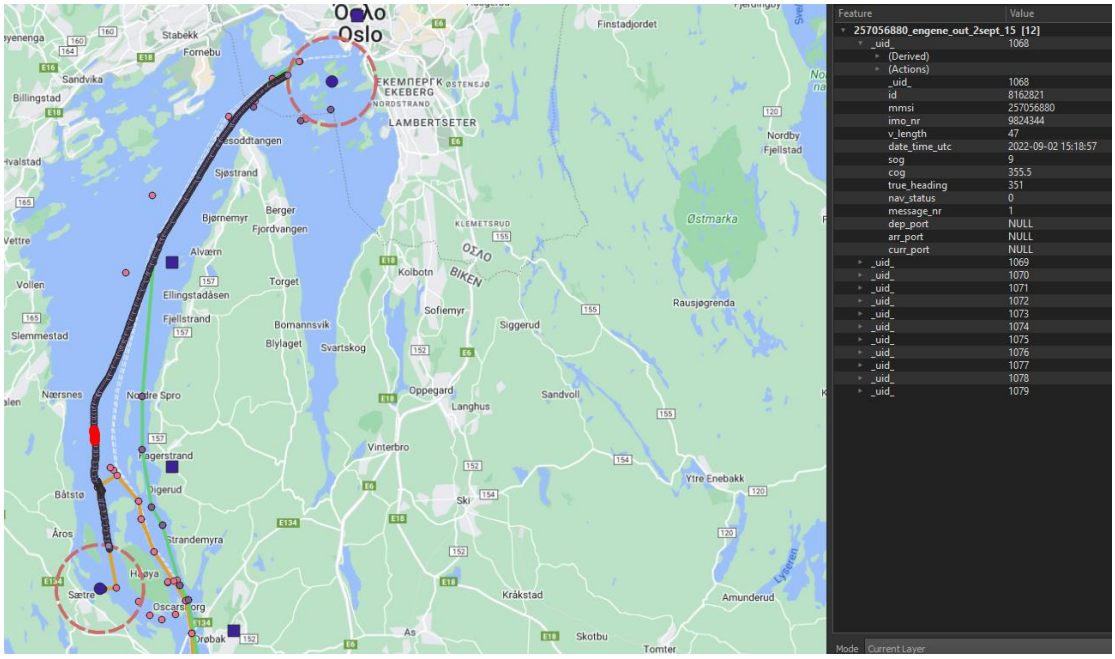
Εικόνα 35: Διαδρομή EngeneSouth_In

start	end	mmsi	trip_id	route_name
2022-09-02 09:05:00	2022-09-02 09:10:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:10:00	2022-09-02 09:15:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:15:00	2022-09-02 09:20:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:20:00	2022-09-02 09:25:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:25:00	2022-09-02 09:30:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:30:00	2022-09-02 09:35:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:35:00	2022-09-02 09:40:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:40:00	2022-09-02 09:45:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:45:00	2022-09-02 09:50:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:50:00	2022-09-02 09:55:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 09:55:00	2022-09-02 10:00:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:00:00	2022-09-02 10:05:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:05:00	2022-09-02 10:10:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:10:00	2022-09-02 10:15:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:15:00	2022-09-02 10:20:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:20:00	2022-09-02 10:25:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:25:00	2022-09-02 10:30:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:30:00	2022-09-02 10:35:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:35:00	2022-09-02 10:40:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:40:00	2022-09-02 10:45:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:45:00	2022-09-02 10:50:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:50:00	2022-09-02 10:55:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 10:55:00	2022-09-02 11:00:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:00:00	2022-09-02 11:05:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:05:00	2022-09-02 11:10:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:10:00	2022-09-02 11:15:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:15:00	2022-09-02 11:20:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:20:00	2022-09-02 11:25:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:25:00	2022-09-02 11:30:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:30:00	2022-09-02 11:35:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:35:00	2022-09-02 11:40:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:40:00	2022-09-02 11:45:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:45:00	2022-09-02 11:50:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:50:00	2022-09-02 11:55:00	257056880	Horten_Engene	EngeneSouth_In
2022-09-02 11:55:00	2022-09-02 12:00:00	257056880	Horten_Engene	EngeneSouth_In

Εικόνα 36: Διαδρομή EngeneSouth_In - αποτελέσματα

5.6 Διαδρομές EngeneNorth_Out και OsloWest_In

Το ίδιο πλοίο, συνεχίζοντας το ταξίδι του Βόρεια, από τις 2022-09-02 12:30:00 έως τις 2022-09-02 16:30:00, εντοπίζεται να ακολουθεί τη διαδρομή EngeneNorth_Out (ροζ διαδρομή), και στη συνέχεια να εισέρχεται στο λιμάνι του Όσλο ακολουθώντας την διαδρομή OsloWest_In, όπως φαίνεται στην Εικόνα 37 και στην Εικόνα 38:



Εικόνα 37: Διαδρομές EngeneNorth_Out και OsloWest_In

start	end	mmsi	trip_id	route_name
2022-09-02 14:55:00	2022-09-02 15:00:00	257056880	Engene_Oslo	EngeneNorth_Out
2022-09-02 15:00:00	2022-09-02 15:05:00	257056880	Engene_Oslo	EngeneNorth_Out
2022-09-02 15:05:00	2022-09-02 15:10:00	257056880	Engene_Oslo	EngeneNorth_Out
2022-09-02 15:10:00	2022-09-02 15:15:00	257056880	Engene_Oslo	EngeneNorth_Out
2022-09-02 15:15:00	2022-09-02 15:20:00	257056880	Engene_Oslo	EngeneNorth_Out
2022-09-02 15:20:00	2022-09-02 15:25:00	257056880	Engene_Oslo	EngeneNorth_Out
2022-09-02 15:25:00	2022-09-02 15:30:00	257056880	Engene_Oslo	OsloWest_In
2022-09-02 15:30:00	2022-09-02 15:35:00	257056880	Engene_Oslo	OsloWest_In
2022-09-02 15:35:00	2022-09-02 15:40:00	257056880	Engene_Oslo	OsloWest_In
2022-09-02 15:40:00	2022-09-02 15:45:00	257056880	Engene_Oslo	OsloWest_In
2022-09-02 15:45:00	2022-09-02 15:50:00	257056880	Engene_Oslo	OsloWest_In

Εικόνα 38: Διαδρομές EngeneNorth_Out και OsloWest_In - αποτελέσματα

6 Συμπεράσματα και μελλοντικές επεκτάσεις

Στην εργασία αυτή ασχοληθήκαμε με το πρόβλημα της εύρεσης του δρομολογίου που είναι περισσότερο πιθανό να ακολουθεί ένα πλοίο καθώς πλέει. Επεξεργαστήκαμε ένα σύνολο δεδομένων από πραγματικά μηνύματα AIS πλοίων που έπλεαν στην περιοχή της Νορβηγίας και μελετήσαμε περιπτώσεις εναλλακτικών δρομολογίων στην ίδια περιοχή. Αναπτύξαμε μια εφαρμογή σε γλώσσα προγραμματισμού Python βασισμένη στην πλατφόρμα παράλληλης επεξεργασία μεγάλων δεδομένων Apache Spark, και συγκεκριμένα στο υποσύστημα Apache Spark Streaming, το οποίο μας επιτρέπει να επεξεργαστούμε τα δεδομένα σε πραγματικό χρόνο. Πειραματιστήκαμε με διάφορα μεγέθη από σύνολα δεδομένων και τιμές σε παραμέτρους και εκτιμήσαμε την απόδοση του συστήματος, μετρώντας το χρόνο εκτέλεσης.

Μέσα από αυτή την εργασία, κατανοήσαμε την υψηλή απόδοση που έχει το Apache Spark στην επεξεργασία μεγάλων δεδομένων, ενώ κατανοήσαμε και την ευκολία κλιμάκωσης που μας παρέχει, σε περίπτωση που οι υπολογιστικές ανάγκες αυξηθούν.

Επίσης, συνειδητοποιήσαμε την αξία που έχει η μελέτη δεδομένων AIS των πλοίων, και τα χρήσιμα συμπεράσματα που μπορούν να εξαχθούν μελετώντας τα τόσο σε χωρικό, όσο και σε χρονικό επίπεδο.

Μία μελλοντική επέκταση που θα μπορούσε να γίνει στη συγκεκριμένη εργασία, θα ήταν να τροποποιηθεί η μετρική βαθμολόγησης των υποψηφίων δρομολογίων με τέτοιο τρόπο ώστε να λαμβάνει υπόψη και την κατεύθυνση με την οποία κινείται το πλοίο, για να προσφέρει μεγαλύτερη ακρίβεια στο ταίριασμα.

Επιπλέον, η ταχύτητα του πλοίου θα μπορούσε να αξιοποιηθεί, ώστε να γίνει μια πρόβλεψη για την ώρα άφιξης του πλοίου στο λιμάνι προορισμού.

Η απόδοση της εφαρμογής θα μπορούσε να βελτιωθεί εκτελώντας την σε μια πραγματική συστάδα κόμβων, και όχι σε ένα τοπικό μηχάνημα που το επίπεδο παραλληλίας που προσφέρει, είναι οι πυρήνες του ίδιο επεξεργαστή.

Τέλος, η ροή δεδομένων θα μπορούσε να προέρχεται από κάποια άλλη υπηρεσία, όπως για παράδειγμα το σύστημα Apache Kafka σε πραγματικό χρόνο, ώστε να τη λαμβάνουμε από εκεί και όχι μέσα από το σύστημα αρχείων.

7 Βιβλιογραφία

[1] Fujino, I., Claramunt, C., & Boudraa, A.-O. (2018). Extracting Courses of Vessels from AIS Data and Real-Time Warning Against Off-Course. In Proceedings of the 2nd International Conference on Big Data Research. ICBDR 2018: 2018 The 2nd International Conference on Big Data Research. ACM. https://doi.org/10.1145/3291801.3291823

[2] Shi, Y., Long, C., Yang, X., & Deng, M. (2022). Abnormal Ship Behavior Detection Based on AIS Data. In Applied Sciences (Vol. 12, Issue 9, p. 4635). MDPI AG. https://doi.org/10.3390/app12094635

[3] Filipiak, D., Węcel, K., Stróżyna, M., Michalak, M., & Abramowicz, W. (2020). Extracting Maritime Traffic Networks from AIS Data Using Evolutionary Algorithm. In Business & Information Systems Engineering (Vol. 62, Issue 5, pp. 435–450). Springer Science and Business Media LLC. https://doi.org/10.1007/s12599-020-00661-0

[4] Automatic Identification System - Wikipedia
https://en.wikipedia.org/wiki/Automatic_identification_system
Πρόσβαση: Φεβρουάριος 2023

[5] Gerard Maas & François Garillot**.** *Stream Processing with Apache Spark: Best Practices for Scaling and Optimizing Apache Spark. Sebastopol, CA*: O'Reilly Media, Inc., 2019

[6] Bill Chambers & Matei Zaharia**.** *Spark: The Definitive Guide: Big Data Processing Made Simple. Sebastopol, CA*: O'Reilly Media, Inc, 2018

[7] Apache Spark – Unified Engine for large-scale data analytics
https://spark.apache.org/
Πρόσβαση: Φεβρουάριος 2023

[8] Distance from a point to a line – Wikipedia
https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line
Πρόσβαση: Φεβρουάριος 2023

[9] Digital route service for navigation
https://www.routeinfo.no/
Πρόσβαση: Φεβρουάριος 2023

[10] AIS Download - Beta
https://ais-public.kystverket.no/ais-download/
Πρόσβαση: Φεβρουάριος 2023

[11] Maritime Safety Information
https://msi.nga.mil/Publications/WPI
Πρόσβαση: Φεβρουάριος 2023

[12] PostgreSQL: The world's most advanced open-source database
https://www.postgresql.org/
Πρόσβαση: Φεβρουάριος 2023

[13] About PostGIS | PostGIS
https://postgis.net/
Πρόσβαση: Φεβρουάριος 2023

[14] QGIS
https://www.qgis.org/en/site/
Πρόσβαση: Φεβρουάριος 2023

[15] Onyango, S.O.; Owiredu, S.A.; Kim, K.-I.; Yoo, S.-L. A Quasi-Intelligent Maritime Route Extraction from AIS Data. Sensors 2022, 22, 8639. https://doi.org/10.3390/s22228639

[16] Dobrkovic, A., Iacob, ME. & van Hillegersberg, J. Maritime pattern extraction and route reconstruction from incomplete AIS data. Int J Data Sci Anal 5, 111–136 (2018). https://doi.org/10.1007/s41060-017-0092-8

[17] Pallotta, Giuliana & Vespe, Michele & Bryan, Karna. (2013). Traffic Route Extraction and Anomaly Detection from AIS Data.

[18] Sheng P, Yin J. Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. Sustainability. 2018; 10(7):2327. https://doi.org/10.3390/su10072327

[19] Sonawane S., Patel D., Kevadiya M., Modi R., Moradiya J., Thomas A. Big Data by 3V's and Its Importance. International Journal of Research in Engineering, Science and Management. Volume-1, Issue-12, December-2018. Pages: 11, 12