



Mälardalen University
School of Innovation, Design and Engineering
Västerås, Sweden

Examensarbete för kandidatexamen i datavetenskap - DVA331 - 15hp

Automatisering av fartygs positionsförutsägelse med hjälp av AIS-data.

Viktor Lund
vld19001
vld19001@student.mdu.se

Examinator: Radu Dobrin
Mälardalens Universitet, Västerås, Sverige

Handledare: Mikael Ekström
Mälardalens Universitet, Västerås, Sverige

Handledare: Jan Gustafsson
Mälardalens Universitet, Västerås, Sverige

Företagshandledare: Andreas Domfors
Qtagg R&D AB, Västerås, Sverige

2023-01-11

Sammanfattning

Denna rapport presenterar ett examensarbete där ett system konstruerats för att undersöka möjligheterna med AIS-data. Automatiskt identifikations system(AIS) har varit ett krav för internationellt gående fartyg i ca 20 år och hur denna data kan användas har undersökts på flera olika sätt. Datan som produceras från AIS innehåller fartygs position, kurs, hastighet, riktning etc. Företaget Qtagg R&D var intresserade av möjligheten att kunna använda AIS-data till förutsägelse av fartygs positioner samt om detta kan utnyttjas för detektering av farliga trafiksituationer. De undrar även om denna process kan automatiseras. Qtagg tror att ett sådant system kan underlätta för ett fartyg under en resa och minska risken för olyckor. Arbeten skedde gemensamt med Qtagg då det fanns ett gemensamt intresse. För att undersöka det valda ämnet insamlades information angående de relevanta områdena, därefter kunde AIS-data lagras med hjälp av en API. Ett system som besvarar frågeställningen konstruerades och testades genom simulering av trafikflödet i den engelska kanalen. Det utfördes 70 simuleringar av två olika typer av resor och frågeställningen kunde besvaras på detta sätt. Vad resultaten visade var att det gick att förutsäga framtida positioner till viss del och att konflikter kunde upptäckas. Genom konstruktionen av systemet blev det möjligt att automatisera processen. Arbetet visar på att utifrån ren AIS-data är det möjligt att automatisera förutsägelsen av fartygs positioner.

Innehåll

1. Inledning	1
2. Bakgrund	2
2.1 AIS-data	2
2.1.1 Räckvidd	2
2.2 Koordinatsystem	2
2.2.1 Longitud och latitud	2
2.3 Incidenter	2
2.4 Databas	3
2.4.1 Relationella databaser	3
2.4.2 SQL	3
2.4.3 PostgreSQL	4
3. Tidigare arbete	5
4. Frågeställning	6
5. Metod	7
6. Etik och samhällliga aspekter	8
7. Implementering	9
7.1 Val av programspråk	9
7.2 Databasens uppbyggnad	9
7.2.1 Datainsamling	10
7.3 Matematiken i systemet	10
7.3.1 Beräkning av fartygs framtida positioner	10
7.3.2 Överlappande trianglar	11
7.3.3 Tidsberäkning	12
7.4 Programuppbyggnad	13
7.4.1 Externa bibliotek	13
7.4.2 Program flöde	13
7.4.3 Pseudokod	16
7.5 Utvärdering av system	16
7.5.1 Testscenarier	17
7.5.2 Testmiljö	17
8. Resultat	18
8.1 Sammanställning	18
9. Diskussion	20
9.1 Analys av resultat	20
9.2 Problem under arbetet	20
9.3 Svar på frågeställningen	20
9.4 Reflektioner	21
10. Slutsatser	22
11. Framtida arbete	23
Referenser	24

Figurer

1	Konceptbild av hur longitud och latitud är definierat, där de horisontella linjerna representerar latitud och de vertikala linjerna representerar longitud.	3
2	Skillnaden mellan en standard enhetscirkel som används inom matematiska utbildningar och den enhetscirkel som används ombord fartyg	11
3	Figuren visar hur trianglar från två olika fartyg kan korsas.	12
4	Flödesdiagram över systemet	14
5	Gränssnittet under körning där potentiella konflikter visas med röda kryss. Dover seaways är det övervakade fartyget och dess triangel är blå, de andra fartygens trianglar är svarta.	15
6	Programmets pseudokod	16

Tabeller

1	Databasens kolumner och deras datatyper	9
2	Olika kombinationer av linjer som kan korsa varandra.	12
3	Sammanställning av resultat	18
4	Resa med mest framgång	18
5	Resa med minst framgång	18
6	Typ 1 resa med mest framgång	18
7	Den sämst presterande resa av typ 2	18

1. Inledning

Inom transportsektorn av handelsvaror är marintransport överlägset störst, ungefär 90% av alla varor i världen sker över vattnet [1]. Hundratusentals fartyg rör sig på vattnet varje dag och det finns risk för fartyg att hamna i potentiellt farliga situationer med andra fartyg. Automatiskt identifikation system eller AIS är ett system som skickar ut positionsdata över radio från fartyg och kuststationer. AIS-data har använts i nästan 20 år. Olika användningsområden där AIS-data kan användas har utforskats i flera sammanhang under åren med fokus på bland annat beräkningar av bränsleförbrukning, beräkning av utsläpp och ruttplanering.

Ett område som också har utforskats är möjligheten att förutsäga ett fartygs position utifrån AIS-data och hur detta i så fall kan implementeras. De arbeten som tidigare utförts inom detta område har antingen kombinerat AIS-data med en eller flera typer av data för att skapa en förutsägelse. Arbeten angående potentiella kollisioner har även utförts med att försöka förutse fartygens positioner. Dessa arbeten har oftast gällt fartyg nära hamnar och för det mesta involverat ett mindre antal fartyg.

Företaget Qtagg i Västerås producerar och installerar olika produkter till stora fartyg såsom motorställdon, ruttplaneringsverktyg och motorregulatorer. De är intresserade av möjligheten att kunna förutse framtida positioner från endast AIS-data och om det går att upptäcka potentiellt farliga situationer. En kund till dem är en färja med en rutt mellan Dover i Storbritannien och Dunkirk i Frankrike. Engelska kanalen är ett geografiskt område där mycket marintrafik rör sig och det är just där detta arbete kommer att fokusera. Målet med detta arbete är att undersöka möjligheten att förutse framtida positioner för fartyg i den engelska kanalen samt hur väl det går utifrån endast AIS-data. De vill också se huruvida det går att upptäcka potentiellt farliga situationer med samma metod och om denna process kan automatiseras. För att besvara dessa frågor konstrueras ett system som simulerar trafikflödet i engelska kanalen och testar de metoder som framtagits i detta arbete.

2. Bakgrund

Denna sektion förklarar och beskriver väsentlig information för att förstå vad arbetet handlar om och vad det försöker uppnå. Olika begrepp och tekniker presenteras samt deras användning och relevans för detta arbete.

2.1 AIS-data

AIS står för Automatiskt Identifikations System och måste enligt *International Maritime Organization*(IMO) [2] användas av alla fartyg med en vikt över 300 ton, men det används även av mindre fartyg frivilligt. AIS har varit ett krav på internationella fartyg sedan januari 2002 och sedan 2005 är alla internationella fartyg utrustade med systemet. Detta system använder sig av två VHF-radiokanaler för att skicka datapaket i synkroniserade tidsintervaller mellan 2 till 10 sekunder. Vad som påverkar intervallet mellan datapaketerna är fartygets hastighet samt dess manövrar. Data som skickas från systemet kallas AIS-data, datan innehåller information angående ett fartygs identitet, position i longitud och latitud, hastighet, kurs etc. Genom ett fartygs AIS-data kan närliggande fartyg upptäckas och presenteras genom olika tekniker där minimikravet är att det visas genom en textdisplay där ID, avstånd och riktning visas på bryggan.

2.1.1 Räckvidd

Räckvidden för AIS-mottagaren ombord beror på flera faktorer såsom om det är en fast eller bärbar mottagare[3]. En bärbar har kortare räckvidd än en fast och kräver ingen VHF-antenn monterad på fartyget. En fast har längre räckvidd och har fler funktioner till exempel inbyggd GPS. Räckvidden kan beräknas på följande vis $D = 2 \sqrt{A}$ där D är räckvidden i nautiska mil och A är antennens höjd över vattnet i meter[3]. Detta betyder att beroende på hur högt över vattnet antennen sitter kommer räckvidden att variera.

2.2 Koordinatsystem

Det primära koordinatsystemet som används idag är WGS84 vilket är framtaget av amerikanska myndigheter och används för bland annat kartografi och GPS-system[4]. WGS84 använder sig av latitud och longitud för att ange koordinater,

2.2.1 Longitud och latitud

Longitud och latitud är ett positionssystem som används över hela världen för att med hög precision ange platser på vår planet[5]. Latitud definierar platsen i nord-sydlig riktning och består av intervallet +90 grader till -90 grader där +90 grader är nordligast och -90 grader är sydligast. Longitud är bredden på jorden och har intervallet +180 grader till -180 grader. +180 grader är längst österut och -180 grader är längst västerut. Longituden är noll grader vid Greenwich strax väster om London, medan latituden är noll grader längs hela ekvatorn. Beroende på vilken grad latitud kommer längden på en grad longitud förändras, detta beror på att vår planet är en sfär. Om det är nära noll grader latitud kommer det skilja väldigt lite på hur lång en grad longitud och en grad latitud är i längd. En grad latitud kommer alltid vara samma längd, medan ju högre eller lägre grad latitud desto kortare kommer en grad longitud vara. Det går att se detta i figur 1 vilket visar konceptet för longitud och latitud.

2.3 Incidenter

Flera hundratusentals fartyg rör sig på haven varje dag [6], det rör sig allt från fiskebåtar, fritidsbåtar till färjor och lastfartyg. Med det höga antalet fartyg uppstår incidenter, antingen mellan flera fartyg eller på grund av andra faktorer såsom miljö eller besättning. Marina olyckor sker inte ofta [7] men även om teknologi och säkerhetsprotokoll förbättras varje år återstår risken för olyckor. Om en incident uppstår kan både kostnader och miljöpåverkan bli väldigt stora. Mellan åren 2011 och 2017 skedde det över 20.000 marina olyckor av olika typer [8], det varierade mellan singelolyckor

Figur 1: Konceptbild av hur longitud och latitud är relaterade, där de horisontella linjerna representerar latitud och de vertikala linjerna representerar longitud.

där en båt var inblandad och incidenter som involverade flera båtar. Resultatet av dessa olyckor var att 203 fartyg antingen sjönk eller blev obrukbara, samt att flera tusen besättningsmedlemmar skadade sig och över 200 personer dog. Även om singelolyckor sker är en krock mellan två fartyg den vanligaste typen av olycka [7]. En krock är vanligast när det kommer till områden där mycket trafik sker till exempel engelska kanalen eller Singaporesundet.

2.4 Databas

En databas är ett verktyg som används i nästan alla program där stora mängder information eller data hanteras.

2.4.1 Relationella databaser

En av de vanligaste typerna av databaser är relationella databaser [9]. En sådan databas strukturerar datan i rader och kolumner i tabeller. Det kan finnas flera tabeller i en relationsdatabas och dessa kan kopplas samman med nycklar. Dessa nycklar är unika identifierare för de existerande relationer som finns i databasen.

2.4.2 SQL

SQL står för Structured Query Language och är det vanligaste programspråket när det kommer till relationella databaser. Språket används för att kommunicera med databasen för att hämta, uppdatera eller införa data. SQL har använts i flera årtionden och flera varianter har framkommit ifrån detta såsom MySQL och PostgreSQL.

2.4.3 PostgreSQL

PostgreSQL är en öppen källkod objekt-relationell databassystem[10] vilket använder SQL tillsammans med era andra funktioner. Databassystemet kan köras på alla stora operativsystem såsom MacOS och Windows. Det har ett bra rykte på grund av sitt skydd av dataintegritet, pålitlighet samt hur det hjälper till att hanterar varierande storlekar av dataset.

3. Tidigare arbete

AIS är ingenting nytt och det finns tidigare arbete där AIS-datans potentiella användningsområden har undersökts. De arbeten som tidigare har skett involverar bland annat användningen av AIS-data för att upptäcka farlig terräng på olika platser samt att kalkylera bränsleförbrukning och utsläpp från ett fartyg.

Bye och Aalberg [11] har granskat olycksrapporter och den korresponderande AIS-datan som var tillgänglig vid olyckor i en statistisk analys. Meningen med deras arbete var att identifiera variabler som påverkar en olycka.

Ett tidigare arbete där AIS-data har använts är en rapport skriven på Wuhans tekniska universitet, där de testade att använda historisk AIS-data vid planering av fartygsrutter i ett område längs Yangtze-oden i Kina där det finns tre raviner och farlig terräng. He et al. [12] föreslår i sin rapport en ny metod för planering av fartygsrutter med hjälp av AIS-data från fartyg som har passerat igenom de områden fartyget kommer passera. De använder den datan för en AI-algoritm med målet att undvika områden där det finns farlig miljö som kan skada fartyget. Deras metod eliminerar den mänskliga faktorn från planering av rutter där personers erfarenhet och känslor påverkar beslut.

Det har även utförts arbeten för att räkna ut fartygs utsläpp samt deras bränsleförbrukning utifrån enbart AIS-data. Ett av dessa arbeten utfördes av Chen et al [13] där de försökte räkna ut den totala mängden utsläpp från fartyg i Tianjin-hamnen i Kina. De undersökte tidigare forskning som fanns relaterat till deras ämnen och försökte sedan skapa en metod som ger ett mer exakt resultat. De lyckades skapa en sådan metod med variabler tagna från AIS-datan samt data från Lloyds databas vilket innehåller information angående fartygs motorer, storlek, vikt och motorens kraft. Genom att kalkylera utsläppen mellan varje sändning av AIS-data och sedan summera detta kan den totala mängden utsläpp beräknas [13]. Metoden visade sig ge ett mer exakt resultat än tidigare använda metoder.

Hur dessa arbeten relaterar till mitt arbete är för att visa hur många olika sätt som AIS-data har använts tidigare och hur väl det har fungerat i tidigare arbeten. De besvarar inte mitt arbetsfrågeställningar då de besvarar frågor angående bränsleförbrukning, utsläppsberäkningar, olycksstatistik och ruttplanering. Den gemensamma faktorn är användandet av AIS-data och att den kan användas på flera olika vis.

4. Frågeställning

Idag vet inte fartygsbesättningar om när eller var en farlig trafiksituation vilken involverar era fartyg kommer att uppstå under resan. Besättningen ombord får istället reagera när en sådan situation uppstår vilket kan medföra stor stress och stora risker. Det här arbetets syfte är att göra det möjligt för besättningen på fartyg att i god tid i förväg få veta var och när en potentiellt farlig situation kan komma att uppstå.

De huvudsakliga frågeställningarna för detta arbete är följande:

- ^ Är det möjligt med hjälp av AIS-data att förutsäga fartygs framtida position i engelska kanalen?
- ^ Kan en möjlig farlig trafiksituation detekteras utifrån den förutsedda positionen?
- ^ Kan denna process automatiseras?

För att besvara dessa frågor kommer ett förutbestämt fartyg i engelska kanalen följas och vi kommer att använda dess insamlade AIS-data för att simulera fartygets färd. Det kommer även insamlas AIS-data för hela engelska kanalen för att kunna simulera ett trafiköde vid testning av programmet.

5. Metod

För att kunna besvara mina frågeställningar behövde det undersökas vad AIS-data är samt möjligheten att få tillgång till dess innehåll. Information angående koordinatsystem behövdes analyseras för att förstå olika aspekter inom fartygsresor. En litteraturstudie utfördes som bestod av relevanta myndigheters webbsidor samt vetenskapliga rapporter angående liknande arbeten inom fältet. Denna information samlas in för att förstå hur AIS-data används samt hur den har utforskats i ett vetenskapligt forskningssammanhang. Matematiken som används i arbetet formulerades efter litteraturstudien.

Systemet kan sedan börja utvecklas, vilket producerar resultatet av arbetet. Den insamlade AIS-data kommer användas vid simuleringar i systemet. Inom simuleringen kommer matematiska formler utnyttjas för att kunna ge en förutsägelse inom vilket område fartyget kan be nna sig i inom en timme samt samt tidsskillnaden av ankomst av två fartyg som en indikator för potentiella kon ikter.

Qtagg i Västerås bidrog med en databas för spara information, AIS-modul samt API-nyckel för att få kunna samla in den relevanta AIS-datan. Dessa komponenter var nödvändiga för arbetets utförande.

6. Etik och samhällliga aspekter

Detta arbete gör inget forskningsetiskt ställningstagande. Arbetet handlar om möjligheterna att använda AIS-data för att förutse framtida positioner av fartyg samt potentiella trafiksituationer.

Dettnas samhällliga aspekter i detta arbete när det kommer till hur det kan bidra till fartygsindustrin. Från ett ekonomiskt samt miljöperspektiv kan detta arbete ses som bidragande. Under ett fartygs resa kan flera ton av bränsle förbrukas och till följd produceras utsläpp vilket påverkar miljön. Med en hög bränsleförbrukning blir detta en stor ekonomisk kostnad för ett fartyg. Genom att förse ett fartyg med mer information kan en mer optimal rutt bli tagen och mindre korrigeringar behöver ske. Information om vart närliggande fartyg befinner sig och deras framtida positioner bidrar till att minska risken för potentiellt farliga trafiksituationer såsom kollisioner eller farliga situationer.

7. Implementering

I denna del beskrivs hur implementationen av systemet har utförts. Det presenteras vilket program-språk som har använts i systemet, vilket operativsystem systemet har testats på, hur databasen är uppbyggd samt hur data har insamlats. Matematiken som används i systemet och vad den bidrar med till systemet presenteras även i detta avsnitt.

7.1 Val av programspråk

Vid valet av programspråk för systemet stod det mellan era objektorienterade språk och i slutet föll valet på språket Python. Anledningen till varför det blev Python är dels på grund av min egna erfarenhet i språket, samt den enkla syntaxen. Python är även ett passande programspråk när det kommer till automatisering för det kan fungera både som ett funktionellt och objektorienterat språk beroende på vad programmet kräver.

7.2 Databasens uppbyggnad

Vid insamling av data har en PostgreSQL databas använts för lagring av AIS-data. Databasen har även använts vid utförandet av simuleringar under systemets testning. All insamlad data sparades i en tabell vilket innehåller följande kolumner med dessa datatyper. Dessa kolumner representerar

Namn	Datatyp
MMSI	varchar
SentTime	Timestamp
Longitude	numeric
Latitude	numeric
CoG	numeric
SoC	numeric
Heading	numeric
RoT	numeric
Navstatus	varchar
ToBow	numeric
ToStern	numeric
ToPort	numeric
ToStarboard	numeric
Draught	numeric
Destination	varchar
RetrievalTime	timestamp
ShipType	varchar
IMO	varchar
ShipName	varchar
ETA	varchar

Tabell 1: Databasens kolumner och deras datatyper

all information som ett AIS-meddelande innehåller och fälten i tabell 1 förklaras nedan.

- ^ MMSI - Maritime mobile service identity identifierar ett fartyg eller kuststation vid radio-kommunikation.
- ^ SentTime - När AIS-meddelande skickades från ett fartyg.
- ^ Longitude - fartygets position i longitud.
- ^ Latitude - fartygets position i latitud.
- ^ CoG - Course over Ground- den kurs fartyget försöker hålla.
- ^ SoC - Speed over Ground- hastigheten fartyget håller när vattenmotstånd inte är medräknat.

-
- ^ Heading - riktningen fartyget håller.
 - ^ RoT - Rate of Turn - hur mycket fartyget svänger i antal grader.
 - ^ NavStatus - vilken status fartyget för närvarande har, den vanligaste statusen är 'Under way, using Engine'. Navigation statusen skickas i numeriska värden från 0 - 15, varje värde har sitt egna förbestämde meddelande. Vid insättning i databasen har dessa värden översatts till de förbestämde meddelande.
 - ^ ToBow, ToStern, ToPort, ToStarboard - avståndet AIS-systemet har till fartygets olika kanter och ger därför systemets position på fartyget.
 - ^ Draught - hur djupt fartyget kommer max vara i vattnet.
 - ^ RetrievalTime - när AIS-datan hämtades genom API:n.
 - ^ ShipType - representerar vilket typ av fartyg det är och använder sig också av ett system där det nns värden mellan 0-99 där varje värde är ett typ av fartyg. Dessa värden översätts till vad de betyder innan de matas in i databasen.
 - ^ IMO - Internationell Maritime Organization identifikationsnummer för fartyg.
 - ^ ShipName - fartygets namn.
 - ^ ETA - Estimated Time of Arrival , när fartygets förväntas anlända till sin destination.

MMSI tillsammans med SentTime utgör primärnyckeln för denna tabell, eftersom MMSI är ett unikt nummer för ett fartyg och i kombination med SentTime blir det en unik identifierare, vilket gör det omöjligt för dubletter att uppstå i den insamlade datan.

7.2.1 Datainsamling

Insamlingen av data skedde med hjälp av en API från AISHub [14]. Det gjordes ett försök till datainsamling med en API från ett annat företag men tyvärr var deras uppdateringsfrekvens av AIS-data för långsamt då det var en timme mellan uppdateringarna. AISHub har realtidsuppdateringar eftersom de samarbetar med olika kuststationer samt varje användare av deras API behöver tillgodose en rå AIS-ström till dem. Med realtidsuppdateringar blir det klarare hur fartygen rör sig samt det blir en större mängd data att använda för systemet och producera ett mer exakt resultat.

Datan hämtas i json-format, sedan görs databasrader av datan och sätts in i databasen. Processen har körts kontinuerligt under era veckor för att kunna ge en stor mängd data att utföra simuleringar på.

7.3 Matematiken i systemet

Matematiken som används i systemet är baserat på användningen av vektorer. I de beräkningar som sker mellan två punkter är antagandet att det är ett ortonometrat koordinatsystem där latitud representeras av Y-axeln och longitud av X-axeln. Anledningen till detta är dels för att underlätta uträkningarna och dels för att planetens sfäriska yta inte har en så stor påverkan för vektorernas längd i detta fall. Ju längre vektorer man arbetar med desto större påverkan kommer planetens sfäriska form påverka beräkningarna.

7.3.1 Beräkning av fartygs framtida positioner

För varje fartyg beräknas två punkter, B och C, som tillsammans med fartygets nuvarande position, A, de bilda en triangel framför fartyget som används för att identifiera olika situationer samt förutspå vart fartyget kommer befinna sig inom den närmaste timmen. I denna sektion förklaras hur de borte hörnen av triangeln punkterna B och C beräknas. För att beräkna positionerna B och C används fartygets position, A, samt fartygets kurs, CoG, och riktning, Heading, vilket ges i grader. Följande beskriver hur punkterna B och C beräknas.

Vanligtvis inom matematik när man använder sig av grader används enhetscirkeln vilket är en cirkel där alla grader existerar. På ett fartyg efterliknas de olika väderstrecken i graderna som kan

Figur 2: Skillnaden mellan en standard enhetscirkel som används inom matematiska utbildningar och den enhetscirkel som används ombord fartyg

ses i gur 2, noll grader är norr, 90 grader är öst, 180 grader är syd och 270 grader är väst. Med detta i åtanke behövs justeringar göras för att få de korrekt cosinus-och sinusvärdena som används för att ta fram punkterna B och C. För att få rätt kurs behöver denna uträkning då göras:

$$= 90$$

Där är den riktning punkten kommer be nna sig i grader och blir då den riktingen i standard enhetscirkeln. Anledningen till varför 90 behöver subtraheras från är för att få vinkelns grader i standard enhetscirkeln. Varför den här konverteringen är nödvändig i systemet är för att få korrekta cosinus- och sinusvärden för hur fartyget rör sig.

Efter har beräknats behöver fartygets hastighetV konverteras från knop till grader i timmen i latitud och longitud. En knop är 1 nautisk mil i timmen, 1 grad latitud är 60 nautiska mil, därför kan konverteringen beräknas genom $V_{lat} = \frac{V}{60}$. Genom dessa två beräkningar är det möjligt att räkna ut hur långt fartyget kommer att färdas på en timme i latitud. Formeln som använts för detta $Y = V_{lat} \sin(\)$. Y är nu skillnaden i grader latitud mellan fartyget och den nya punkten. Den nya punktens latitud kan sedan nnas genom $q_y = A_y + Y$. Longituden för den nya punkten kan beräknas på nästan samma vis. Beroende på graden latitud kommer längden för 1 grad longitud variera, se konceptet i gur 1. Vid 0 grader latitud är 1 grad longitud 60 nautiska mil, och när antalet grader latitud ökar eller minskar förändras antalet nautiska mil per grad longitud på grund av att vår planet är en sfär. Därför behövs denna formel använda $D = 60 \cos(A_y)$. D är antalet nautiska mil för en grad i longitud vid A_y grader latitud. Sedan kan fartygets hastighet i longitudled beräknas med $V_{long} = \frac{V}{D}$. Nu kan sträckan fartyget färdas i longitudled under en timme beräknas, genom formeln $X = V_{long} \cos(\)$. Efter att X har beräknats kan q_x bestämmas med formeln $q_x = A_x + X$.

q_y och q_x är då koordinaterna för en av de nya punkterna som utgör en av de tre hörnen av den triangel inom vilken fartyget förutses komma be nna sig inom den närmaste timmen. Framtagandet q_y och q_x är då koordinaterna för en av de nya punkterna som utgör en av de tre hörnen av den triangel inom vilken fartyget förutses komma be nna sig inom den närmaste timmen. Framtagandet av punkterna sker för varje fartygs punkt B och C för att skapa triangelns två bortre hörn.

7.3.2 Överlappande trianglar

För att upptäcka potentiella tra ksituationer behöver systemet kontrollera om fartygs trianglar skär varandra och i så fall var de skär varandra, konceptet kan ses i gur 3. Det är tre steg som genomförs för att se om två linjer skär varandra. Steg ett är att se om linjerna är parallella eller inte. Formeln för att att se om de är parallella är

$$T = (L2_{y2} \ L2_{y1}) \ (L1_{x2} \ L1_{x1}) \ (L2_{x2} \ L_{x1}) \ (L1_{y2} \ L1_{y1})$$

där L1 och L2 är de två linjerna och L_{1x}, L_{1y} representerar en ände av linjen, detta gäller för båda linjerna. Om L1 och L2 är parallella kommer T vara lika med 0 och det betyder att de inte korsar varandra. Om linjerna inte är parallella behöver det kollas om linjerna korsar varandra. För att linjerna ska korsa varandra, och inte är bortom varandras räckvidd, behövs det att U_a och U_b 2 (0; 1), där U_a och U_b beräknas med hjälp av formlerna:

$$U_a = \frac{(L_{2x2} - L_{2x1}) (L_{1y1} - L_{2y1}) - (L_{2y2} - L_{2y1}) (L_{1x1} - L_{2x2})}{T}$$

och

$$U_b = \frac{(L_{1x2} - L_{1x1}) (L_{1y1} - L_{2y1}) - (L_{1y2} - L_{1y1}) (L_{1x1} - L_{2x1})}{T}$$

Om linjerna korsar varandra beräknas koordinaterna för skärningspunkten. För longitud, eller X koordinaten, används följande formel $S_x = x_1 + U_a (x_2 - x_1)$ där S_x är skärningspunktens X-koordinat. Skärningspunktens Y-koordinat bestäms med hjälp av formeln $S_y = y_1 + U_a (y_2 - y_1)$. Dessa beräkningar görs för de nio olika kombinationer som linjerna kan korsa varandra.

Om en triangel har hörnen A, B och C nns kanterna AB, AC och BC. Två trianglars sidor kan då korsa varandra i de kombinationer som kan ses i tabell 2. Ett exempel på två trianglar som korsar varandra kan ses i gur 3.

Linje 1	Linje 2
A ₁ B ₁	A ₂ B ₂
A ₁ B ₁	A ₂ C ₂
A ₁ B ₁	B ₂ C ₂
A ₁ C ₁	A ₂ B ₂
A ₁ C ₁	A ₂ C ₂
A ₁ C ₁	B ₂ C ₂
B ₁ C ₁	A ₂ B ₂
B ₁ C ₁	A ₂ C ₂
B ₁ C ₁	B ₂ C ₂

Tabell 2: Olika kombinationer av linjer som kan korsa varandra.

Figur 3: Figuren visar hur trianglar från två olika fartyg kan korsas.

7.3.3 Tidsberäkning

Om en potentiell koniktzon har upptäckts av systemet kommer det ske beräkningar dels för hur lång tid det är kvar till dess det egna fartyget når zonen, men även tidsskillnaden mellan när de två fartygen når zonen.

För att bestämma hur lång tid det är kvar till dess det egna fartyget når en koniktzon behövs först avståndet till zonen beräknas. Vid beräkning av distansen till zonen bestäms avståndet till

varje hörn av zonen, detta görs för att skapa ett intervall utav distanser till varje hörn av zonen. Från detta intervall går det sedan att räkna ut den ungefärliga tiden till när en konikt potentiellt kan ske. När avståndet till ett hörn beräknas används formeln för Euklidiskt avstånd:

$$\text{Distans} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (1)$$

Efter att avståndet har bestämts för ett hörn av zonen kan tiden till dess det egna fartyget når dit beräknas enligt formeln: $Tid = \frac{\text{Distans}}{\text{Fart}}$,

då vi vet distans behövs hastigheten bestämmas. Fartygets hastighet anges i knop, därför behöver det konverteras till hastighet i grader latitud och longitud. Formeln för konverteringen:

$$V_{\text{lat}} = \frac{(\text{SoG} \cdot \sin(\text{Heading}))}{60}$$

och

$$V_{\text{long}} = \frac{(\text{SoG} \cdot \cos(\text{Heading}))}{(60 \cdot \cos(A_y))}$$

farten är då

$$\text{Fart} = \sqrt{V_{\text{lat}}^2 + V_{\text{long}}^2}$$

där farterna V_{lat} och V_{long} är latitud- och longitudgrader per timme. A_y är fartygets position i latitud och är nödvändig att inkludera i denna uträkning då som tidigare sagt en grad longitud är inte alltid 60n nautiska mil per grad. Därefter kan tiden tills fartyget når hörnet beräknas genom distans dividerat med hastighet.

Samma beräkningar sker sedan för den andra deltagaren i den potentiella konikten. Efter att båda tiderna har räknats ut kan tidsskillnaden beräknas genom $t = t_2 - t_1$, t_1 är tiden för det egna fartyget och t_2 är tiden för det andra fartyget. Eftersom dessa beräkningar görs för varje hörn skapas även här ett intervall men i detta fall av tidsskillnader. Systemet har nu beräknat den minsta tidsskillnaden i ankomst för de bägge fartygen samt den största skillnaden. Detta intervall blir sedan presenterat till användaren i gränssnittet.

7.4 Programuppbyggnad

I denna sektion beskrivs vilka externa bibliotek som har använts för att skapa programmet, hur systemet har byggts upp samt programmets öde.

7.4.1 Externa bibliotek

Det har använts era externa bibliotek i implementationen av systemet för att hjälpa till med olika funktionaliteter. Dessa externa bibliotek underlättar de beräkningar som behöver göras i systemet. Ett gra skt gränssnitt implementeras med hjälp av ett av dessa bibliotek för att öka användarvänligheten och ge en visuell bekräftelse om händelser i systemet.

De externa biblioteken är:

`Tkinter` : Detta bibliotek används till det gra ska gränssnittet.

`TkinterMapView` : Tillför en interaktiv karta till gränssnittet. Gör det möjligt att markera ut GPS-positioner.

`Psycopg2` : Används för att ansluta samt kommunicera med databasen.

`Shapely` : Tillför geometriska funktioner.

7.4.2 Program öde

I gur 4 går det att se systemets öde. Det börjar med att upprätta en anslutning till databasen, därefter hämtar systemet den insamlade datan för att simulera att AIS-data hämtas från andra

Figur 4: Flödesdiagram över systemet

fartyg. Nästa steg är att beräkna alla fartygs trianglar för att förutspå fartygens framtida positioner. Därefter körs funktionaliteten som beräknar hur framgångsfull en förutsägelse var. Systemet kontrollerar om det finns några potentiella kon ikter för det övervakade fartyget. Avståndet beräknas till alla kon iktpartners som sedan sparas för att se om avståndet har krympt nästa iteration. För potentiella kon iktpartners behöver det sedan beräknas vart kon iktzonerna befinner sig, tiden tills fartygen når zonerna samt tidsskillnaden i ankomst till zonerna. Därefter kategoriseras de potentiella kon ikter baserat på hur mycket tid det är tills det egna fartyget når zonen samt hur stor tidsskillnaden tills de båda fartygen ankommer till zonen. Till sist ritas alla kon iktpartners och deras trianglar upp i gränssnittet tillsammans med det egna fartyget samt en lista med möjliga kon ikter, mmsi-nummer, tid tills zonen nås och tidsskillnad i ankomst. Figur 5 visar gränssnittet under en simulering.

Figur 5: Gränssnittet under körning där potentiella kon ikter visas med röda kryss. Dover seaways är det övervakade fartyget och dess triangel är blå, de andra fartygens trianglar är svarta.

7.4.3 Pseudokod

Figur 6: Programmets pseudokod

I figur 6 kan pseudokoden för programmet ses.

7.5 Utvärdering av system

Under testningen fokuserades hur väl systemet kan förutse fartygs framtida positioner. Detta på grund av det är systemets huvudfunktion och att det är där Qtaggs största intresse ligger. Det som även testas är hur många potentiella kon ikter som påträ as under en resa samt hur nära de andra fartygen kommer till det övervakade fartyget.

Anledningen till varför förutsägelsen av framtida positioner testas är för att se hur väl systemet kan förutse det egna fartygets position. Testet för detta var från tiden när ett AIS-meddelande kommer kontrolleras varje AIS-meddelande för fartyget om nästa position be nner sig i triangeln. För varje ny position kommer en ny triangel ritas ut. Om en ny position be nner sig i en triangel kommer 1 adderas till en variabel vars syfte är att hålla räkning på hur många minuter de framtida positionerna be nner sig i en triangel. Den kortaste tiden ett fartyg kan be nna sig i en triangel är 0, vilket betyder att nästa position aldrig befann sig i triangeln. Den längsta tiden är 60 minuter vilket betyder att systemet kunde förutse 100% av de framtida positionerna. Från detta blir det sedan en relativ andel i procent av hur mycket av den timmen systemet lyckades förutse. Alla trianglar som skapats under detta test får en procentsats över hur väl framtida positioner förutspåddes från den punkten.

Genomsnittet för hur lång tid en resa tar mäts för att se om det är en skillnad beroende på längden

av resan. Eftersom det är simuleringar är tidsgränserna redan satta beroende på klockslagen första meddelandet och sista meddelandet skickades. Det går då att se om längre eller kortare resor har en tendens att ha mindre eller högre procent gällande förutsägelse.

Hur nära fartyg kom till det egna fartyget testades genom att varje omgång testa om en ny potentiell konikt har förekommit och notera avståndet till det fartyget. Om ett fartyg redan har noterats kommer avståndet testas för att se om distansen har minskat mellan fartygen. Anledningen till varför detta testas är för att se hur nära fartyg kommer och se om avstånden kan påverka förutsägelsen.

Till sist ses noteras hur många potentiella koniktpartners som påträffas under resan och dess påverkan på hur väl de framtida positionerna förutspås. Ett fartyg kan behöva ändra sin riktning eller kurs på grund av mycket trafik i närheten och det kan påverka hur väl de framtida positionerna förutspås.

7.5.1 Testscenarier

Under testningen är det två typer av scenarier som testades. Typ 1 är resor som startar i en hamn i engelska kanalen och avslutas i en annan hamn i engelska kanalen. Typ 2 är resor som inte startar eller avslutas i den engelska kanalen utan bara korsar genom den övervakade zonen. Varför dessa två typer testas är för att se om början och slutet påverkar hur väl det går att förutse positioner.

Totalt testades 70 unika resor i systemet, antalet resor ansågs vara en passande provstorlek i relation till den mängd AIS-data som insamlats under arbetet. Av de resorna var 50 resor mellan Dover och Dunkirk och faller inom resor av typ 1, vilket betyder att de har kunnat övervakas under hela sin resa. Av de 50 resorna är 25 resor från Dover till Dunkirk och 25 är omvända. De andra 20 resorna är gjorda av fartyg som korsar zonen och är då av typ 2. De 20 resorna som valdes ut började sin resa utanför den övervakade zonen och avslutade även sin resa utanför zonen. Restiden för resorna togs i hänsyn med kravet att resorna inte kunde vara kortare än en timme då längden på resan ansågs vara för kort.

7.5.2 Testmiljö

Den testmiljö som har använts vid framställande av resultat har följande specifikationer:

Processor: Intel Core i5-9300H
Ram: 8GB 2400mhz
Grafik: NVIDIA GeForce GTX 1050
Minne: 256GB HFM256GDHTNG-8310A
Trådlöst nätverk: Intel Wireless-AC 9462
Operativsystem: Windows 11 Home
Version: 21H2
OS-version: 22000.1098
Kompilator: Python 3.10.

Anslutning till databasen sker genom Qtags trådlösa nätverk. Maskinen där databasen befinns är ansluten till nätverket genom Ethernet-kabel. Systemet körde i debugläge i visual studio code under testerna. Det fanns ingen speciell anledning varför systemet körde i detta läge under testerna.

8. Resultat

I denna sektion presenteras och analyseras resultaten från testerna.

8.1 Sammanställning

Typ av resa	Lyckad förutsägelse	Medel restid	Konflikter funna	Medel avstånd
Typ 1	16%	2h 17min	88	4.44NM
Typ 2	32%	4h 5min	58	1.56NM
Medelvärde	21%	2h 47min	80	3.55

Tabell 3: Sammanställning av resultat

I tabell 3 går det att se sammanställningen av resultaten. Resultaten består av hur väl framtida positioner har lyckats, hur lång medelrestiden var för resorna, hur många konflikter som uppstod i genomsnitt, samt vad genomsnittet på det minsta avståndet mellan det övervakade fartyget och andra fartyg. För resor av typ 1 var andelen av lyckade förutspådda framtida positioner 16%, medan för resor av typ 2 var det 32%. Det var skillnad i snittrestid för de två typerna av resor där korsande fartyg hade 1h 48min längre restid än resor av typ 1. Korsande fartyg hade i genomsnitt 30 mindre funna konflikter och potentiella konflikt partners befann sig i genomsnitt 2.88 nautiska mil¹ närmare det övervakade fartyget än i resor av typ 2.

Den resa med mest framgång gällande förutsägelse av framtida positioner var ett fartyg som korsade zonen. Som kan ses i tabell 4 under resan kunde 48% av fartygets framtida positioner förutspås.

Lyckad förutsägelse	Restid	Konflikter funna	Medel avstånd
48%	3h 45min	43	1.13NM

Tabell 4: Resa med mest framgång

Resan hade nästan en timme längre restid än genomsnittet. När det kommer till antalet konflikter funna samt hur nära andra fartyg kom till det övervakade fartyget faller denna resa under genomsnittet i dessa kategorier. Den minst framgångsfulla prestationen kom under en resa av typ 1. Resan som presenteras i tabell 5 hade en kortare restid än genomsnittet, fler funna konflikter samt

Lyckad förutsägelse	Restid	Konflikter funna	Medel avstånd
3%	2h 19min	111	4.63NM

Tabell 5: Resa med minst framgång

längre avstånd till potentiella konflikt partners.

För resorna av typ 1 har den minst lyckade resan presenterats ovan. Den mest framgångsrika resan av typ 1 kan ses i tabell 6. Med 28% hade den högst framgång när det kommer till förutsägelse av

Lyckad förutsägelse	Restid	Konflikter funna	Medel avstånd
28%	2h 21min	69	3.2NM

Tabell 6: Typ 1 resa med mest framgång

Lyckad förutsägelse	Restid	Konflikter funna	Medel avstånd
16%	2h 22min	52	2.16NM

Tabell 7: Den sämst presterande resa av typ 2

framtida position för resor av typ 1. I de tre övriga kategorierna var resan under medelvärdet. Det

bästa resultatet från resor som korsar zonen visas i tabell 4. För restypen kan den sämsta resan ses i tabell 7, vilket visar en 16% lycka när det kommer till förutsägelse av framtida positioner. Restiden är under genomsnittet, konflikter funna under resan är också under genomsnittet, det minsta avståndet till konflikt partners är också under genomsnittet.

9. Diskussion

Vad som presenteras i denna sektion är analys av resultaten, reflektion av arbetet samt diskussion angående hur frågeställningen har besvarats.

9.1 Analys av resultat

Från resultaten ses att förmågan att förutse framtida positioner är relativt låg för resor av typ 1, medan däremot förmågan i är genomsnitt dubbel så hög för resor av typ 2. Vad som skiljer dessa två typer av resor är att typ 2 gäller båtar som redan befinner sig i en resa och behöver inte ta sig ut från en hamn eller förankra i en. Detta verkar vara en faktor som ökar andelen korrekta förutsägelser av framtida positioner. När den resa med mest framgång och resan med minst framgång jämförs ser man att vad som skiljer dem åt är att de ligger på olika sidor av medelvärdet i alla kategorier. När det kommer till restid har den mest framgångsrika 58 minuter längre restid än genomsnittet medan den minst lyckade resan har 28 minuter kortare än genomsnittet, vilket kan betyda att ju längre resa desto mer tid spenderas på samma kurs och färre manövrar görs. Om fokuset faller på antalet konflikter funna under resan ser vi att ungefär hälften av genomsnittet blev funna under den bäst presterande resan, medan under den sämst presterande resan var antalet 111 konflikter vilket är 39% över genomsnittet. Detta kan tyda på när många potentiella konflikter är funna påverkar det systemets förmåga att förutse fartygets framtida positioner på grund av de manövrar fartyget behöver göra för att undvika de andra fartygen. Färre konflikter kan då betyda att en högre grad av framgång kan uppnås gällande förutsägelse av framtida positioner. Detta stärks även genom att se på tabell 6 där antalet konflikter också faller under genomsnittet.

9.2 Problem under arbetet

Det största problemet under arbetet var den tid som spenderades på den först tilltänkta API:n. Under 1,5 månader användes den API:n innan jag upptäckte att AIS-datan uppdaterades med en timmes mellanrum. Det långa intervallet skulle göra det svårt att kunna köra simuleringar effektivt då trafikflödet inte skulle kunna följas lika noggrant. Efter att intervallet upptäcktes rekommenderade min handledare från Qtagg en API som han hade upptäckt och efter lite efterforskningar börjades AIS-data insamlas med hjälp av AIShub API:n [14]. I och med denna miss vid mitt val av API påbörjades datainsamlingen något senare än tänkt, men det påverkade inte hur många simuleringar som kunde utföras under testerna.

9.3 Svar på frågeställningen

De frågor som arbetet skulle besvara var följande:

- Är det möjligt med hjälp av AIS-data att förutsäga fartygs framtida position i engelska kanalen?
- Kan en möjlig farlig trafiksituation detekteras utifrån den förutsedda positionen?
- Kan denna process automatiseras?

Först ser vi på den första frågan om det går att förutse fartygets positioner utifrån AIS-data. Vad som besvarades gällande den frågan är om framtida positioner kunde förutses med den metod som använts i arbetet. Resultatet visar att det går att förutse positioner med denna metod med viss framgång. Frågan har då besvarats att det är möjligt att förutse ett fartygs position. Hur väl positionerna kunde förutses varierar för varje resa och var i resan fartyget befinner sig.

Den andra frågan i min frågeställning om en möjligtvis farlig situation kan upptäckas utifrån en förutsedd position har även besvarats. Potentiella konflikter kunde upptäckas under simuleringen, men utan data från riktiga konflikter är det svårt att avgöra hur väl konflikterna skulle kunna förhindras med denna metod. Utifrån resultaten kan man se att ett stort antal potentiellt farliga situationer blev funna av systemet, vilket betyder att frågan har besvarats med att det är möjligt.

Den sista frågan av frågeställningen rörande om processen kan automatiseras svarades på genom att konstruera det system som gjorde det möjligt att besvara de andra två frågorna i min frågeställning. Systemets olika funktionaliteter tillät automationen av processen genom hämtning av AIS-data, de beräkningar systemet utför samt de konstanta uppdateringar programmet gör.

9.4 Reflektioner

Det första jag vill reflektera över är de begränsningar som sattes för arbetet, de var att bara ett fartyg skulle följas och simuleras. Det andra var att det endast var AIS-data för trafiken i den engelska kanalen som skulle insamlas och användas i simuleringarna. Efter att ha utfört arbetet och utvärderat dessa begränsningars påverkan kan jag säga att de var passande för storleken av arbetet. Hade arbetet haft mindre begränsningar skulle nog arbetet varit för stort för att kunna slutföras i den tidsram som givits.

När det kommer till hur detta arbete kan relateras till de tidigare arbeten som nämnts i rapporten kan kopplingar göras till hur de kan subsidiera varandra. Det tidigare arbete som kommer till åtanke är He et al.[12] arbete angående användandet av historiskt AIS-data vid ruttplanering. Hur dessa två arbeten kan subsidiera varandra är genom en kombination av de två där ett fartyg använder He et al. metod för att planera resan för att undvika farlig miljö och sedan metoden för detta arbete under resan för att undvika farliga trafiksituationer. Dessa två arbeten skulle nog fungera bra tillsammans i områden där det är mycket trafik och det finns risk för att gå på grund eller skada fartyget. Ett exempel skulle vara floden Yangtze i Kina som har varit en viktig handelsled i hundratals år då floden rinner igenom Kina ut till Stilla havet. Det är många fartyg i floden och stora fartyg kan ha det svårt att manövrera i potentiella konflikter. Kombinationen av dessa två arbeten skulle då kunna underlätta för ett fartyg genom att tillhandahålla mer information och göra det enklare att behålla kursen och påvisa vilka manövrar som kan behöva utföras.

10. Slutsatser

Målet med arbetet var att använda AIS-data för att förutse fartygs positioner i engelska kanalen, om potentiellt farliga trafiksituationer kan upptäckas och om denna process kan automatiseras. För att besvara dessa frågor samlades information angående AIS-data samt de relevanta områdena inom marinnavigering in och användes till formulering av matematiska formler.

Ett system skapades för att undersöka frågeställningen. En datainsamling upprättades i syfte att använda den insamlade AIS-datan för att simulera trafikflödet i den engelska kanalen och under simuleringarna testa den framtagna metoden. Vad som testades var hur väl systemet kunde förutse ett fartyg under en resa, hur många potentiella konflikter som påträffades, minsta avståndet till andra fartyg samt hur lång tid resan tog. Resultaten från testerna visade att det är möjligt att förutse positioner utifrån AIS-data med denna metod men med relativt låg framgång. Under en resa kunde framtida positioner förutses 20% av tiden inom den kommande timmen. Efter analys av resultaten var det möjligt att se om de andra faktorerna som testades hade någon påverkan på systemets förmåga att förutse positioner. Det visade sig att det påverkade hur framgångsfullt systemets förutsägelser var. Systemet bevisade att potentiella konflikter var möjliga att upptäcka med den använda metoden och ett flertal blev funna under varje simulering.

11. Framtida arbete

Om arbetet skulle fortsätta vore ett intressant tillvägagångsätt vara att försöka implementera en modell som utnyttjar artificiell intelligens för att förutse framtida positioner. En AI-modell som är intressant för detta skulle vara *Ant Colony Optimization* även kallat ACO. Det är en optimerings-algoritm och har använts vid till exempel val av fordonsrutter. Ett sätt att implementera ACO skulle då vara att applicera ett rutnät av celler över den sträcka resan kommer gå. En metod för att ta reda på andra fartygs destinationer behöver utarbetas. I nuläget finns det inte en standard för hur en destination ska skrivas i en sändning av AIS-data och därför kan det variera väldigt mycket från fartyg till fartyg. Sedan skulle ACO appliceras där myrornas startposition skulle vara ett fartyg i trafiken och målet är fartygets destination. Cellerna skulle sedan fyllas med feromoner där de celler som har besökts mest av fartyg får ett högre antal feromoner för att visa var fartyg oftast åker. Algoritmen skulle appliceras på varje fartyg som möts under resan. Vad som skulle kunna vara ett problem för denna metod är hur väl den skulle kunna prestera när det är många fartyg behöver kontrolleras. Om det egna fartyget möter 50 fartyg under en iteration skulle det behövas 50 förfrågningar till databasen, 50 feromon-initieringar och sedan behöver myrorna iterera tills den mest troliga rutten har blivit funnen. Exekveringen av bara en omgång av ACO kan då ta lite tid, jag vet eftersom jag har egen erfarenhet av att implementera algoritmen. Om en parallell exekvering skulle implementeras kan det minska den tid som skulle behövas för att utföra algoritmen på alla möta fartyg. Det är fortfarande en fråga om hur effektivt det skulle vara samt att rätt mycket tid måste allokeras för att göra en algoritm som kan avläsa ett fartygs destination i en sändning av AIS-data. Avläsning av detta är nog bland det mest tidskrävande då som tidigare nämnt varierar mycket beroende på fartyg. En stor insamling av AIS-data behöver utföras för att kunna implementera ACO och producera ett svar med relativt hög framgång. ACO är nog den väg jag skulle välja vid en fortsättning av detta arbete för det är en passande modell som har redan testats till viss del av He et al. [12] och personligen anser jag att det är fullt möjligt att applicera ACO i en fortsättning av detta arbete.

En fortsättning på konfliktdetekteringen skulle vara en *Velocity Obstacle* eller VO-modell som implementeras istället för den triangel som använts i detta arbete. Det är en modell som används framförallt inom robotik. En implementation skulle kunna vara en förbättring eftersom det är en metod som har blivit prövad ett flertal gånger.

Referenser

- [1] W. Zhang, F. Goerlandt, J. Montewka och P. Kujalam, "A method for detecting possible near miss ship collisions from AIS data," *Ocean Engineering*, vol. 107, ss. 60-69, Aug. 2015, DOI: 10.1016/j.oceaneng.2015.07.046.
- [2] Sjöfartsverket, "Frågor och svar om AIS-data". 2021. [Online]. Tillgänglig: <https://www.sjofartsverket.se/sv/tjanster/bygg-och-teknik/fragor-och-svar-om-ais-data/> (hämtad: 2022-09-07).
- [3] Marinwebben, "Information om VHF". 2020, [Online]. Tillgänglig: <https://www.marinwebben.se/vhf> (hämtad: 2022-09-08).
- [4] Lantmäteriet, "WGS84". 2022. [Online]. Tillgänglig: <https://www.lantmateriet.se/sv/geodata/gps-geodesi-och-swepos/Referenssystem/Tredimensionella-system/WGS-84/> (hämtad: 2022-10-11).
- [5] Britannica, "Latitude and longitude". 2022. [Online]. Tillgänglig: <https://www.britannica.com/science/Latitude>. (hämtad: 2022-11-01).
- [6] MarineTraffic, "Live Map". 2022, [Online]. Tillgänglig: <https://www.marinetraffic.com/en/ais/home/centerx:12.1/centery:50.5/zoom:6> (Hämtad: 2022-09-03).
- [7] W. Zhang, F. Goerlandt, P. Kujalam och Y. Wang, "An advanced method for detecting possible near miss ship collisions from AIS data," *Ocean engineering*, vol. 124, ss. 141-156, Aug. 2016, DOI: 10.1016/j.oceaneng.2016.07.059.
- [8] S. Yildiz, Ö. Uğurlu, J. Wang och S. Loughney, "Application of the HFACS-PV approach for identification of human and organizational factors (HOFs) influencing marine accidents," *Reliability engineering system safety*, vol. 208, ss. 107395, Dec. 2020, DOI: 10.1016/j.res.2020.107395.
- [9] IBM, "Relational Databases". 2019, [Online]. Tillgänglig: <https://www.ibm.com/cloud/learn/relational-databases> (Hämtad: 2022-10-26).
- [10] "About". 2020, [Online]. Tillgänglig: <https://www.postgresql.org/about/> (Hämtad: 2022-10-22).
- [11] R. J. Bye och A. L. Aalberg, "Maritime navigation accidents and risk indicators: An exploratory statistical analysis using AIS data and accident reports," *Reliability engineering & system safety*, vol. 176, ss. 174-186, Aug. 2018, DOI: 10.1016/j.res.2018.03.033.
- [12] Y. K. He, D. Zhang, J. Zhang, M. Zhang och T. W. Li, "Ship Route Planning Using Historical Trajectories Derived from AIS Data," *TransNav*, vol. 13, nr. 1, Mar. 2019, DOI: 10.12716/1001.13.01.06.
- [13] D. Chen et al., "Estimating ship emissions based on AIS data for port of Tianjin, China," *Atmospheric Environment*, vol. 145, ss. 10-18, Nov. 2016, DOI: 10.1016/j.atmosenv.2016.08.086.
- [14] AISHub, "AIS data API". 2022. [Online]. Tillgänglig: <https://www.aishub.net/api> (Hämtad 2022-10-23).