National Technical University of Athens

Diploma Thesis

"Development of a Negative Selection Algorithm for Ship Collision Risk Detection"

Konstantinos Giakoumakis

Supervisors:

Nikolaos P. Ventikos

Athens, 2023

Acknowledgement

This thesis represents the culmination of my studies in the School of Naval Architecture and Marine Engineering at the National Technical University of Athens, and has been conducted under the guidance of the Ship Design and Maritime Transport sector.

At this point, I would like to express my heartfelt gratitude to my thesis advisor, Professor Nikolaos Ventikos, for his invaluable guidance, support and encouragement throughout my research journey. His expertise, insightful comments and constructive criticisms have been instrumental in shaping my ideas and developing my skills.

Furthermore, I am deeply grateful to Mr. Konstantinos Louzis for his invaluable advice and guidance throughout the process of writing this thesis. I am thankful for his encouragement, insightful feedback, and support which greatly contributed to the successful completion of this thesis.

I would like to extend my thanks to my employer, Angelicoussis Shipping Group, for granting me the time and resources to pursue this research. Their support has been a great motivation to me and has helped me to balance my professional responsibilities with my academic pursuits.

Finally, I would like to thank my family and friends for their unwavering love, encouragement and support. Without their emotional support and understanding, I would not have been able to complete this project.

Περίληψη

Η σύγκρουση μεταξύ πλοίων είναι αδιαμφισβήτητα ένα από τα θαλάσσια ατυχήματα με τις πιο σημαντικές επιπτώσεις. Η έρευνα για την ανάπτυξη μεθόδων πρόληψης συγκρούσεων λοιπόν είναι επιτακτική για την ασφάλεια τόσο του θαλάσσιου περιβάλλοντος όσο και των ναυτικών και των επιβατών που εμπλέκονται στις θαλάσσιες μετακινήσεις. Οι ερευνητές έχουν αναπτύξει μεθόδους πρόληψης των συγκρούσεων που μπορούν να ταξινομηθούν σε γενικές γραμμές στην ανίχνευση του κινδύνου της σύγκρουσης (collision risk detection) και στην αποφυγή της σύγκρουσης (collision avoidance). Η ανίχνευση του κινδύνου της σύγκρουσης αποσκοπεί στην πρόβλεψη μιας επικείμενης σύγκρουσης δίνοντας το έναυσμα για την αποφυγή της σύγκρουσης η οποία περιλαμβάνει τον καθορισμό των απαραίτητων ενεργειών για την αποφυγή του ανιχνευμένου κινδύνου. Στην παρούσα διπλωματική εργασία επικεντρωθήκαμε στην ανάπτυξη μιας νέας μεθόδου ανίχνευσης του κινδύνου της σύγκρουσης. Η μέθοδος μας είναι ένας αλγόριθμος machine learning και πιο συγκεκριμένα ένας αλγόριθμος αρνητικής επιλογής (negative selection algorithm).

Προκειμένου να επιτύχουμε τον στόχο της ανάπτυξης αυτού του αλγορίθμου, αξιοποιήσαμε γνώσεις τόσο από την έρευνα της ανίχνευσης κινδύνου σύγκρουσης πλοίων όσο και από τον τομέα των Artificial Immune Systems, τα οποία περιλαμβάνουν και το Negative Selection Algorithm. Τα αρχικά στάδια της εργασίας μας επικεντρώθηκαν στην ανασκόπηση και τη σύνθεση της υπάρχουσας έρευνας σε αυτούς τους τομείς, εντοπίζοντας βασικές γνώσεις που ενέπνευσαν την προσέγγισή μας. Στη συνέχεια, αναπτύξαμε έναν δευτερεύοντα αλγόριθμο με σκοπό να κατασκευάσουμε τα δεδομένα που απαιτούνται για την εκπαίδευση και την αξιολόγηση του αλγορίθμου αρνητικής επιλογής. Εφόσον εξασφαλίσαμε τα απαραίτητα δεδομένα, προχωρήσαμε στη συνέχεια στην εκπαίδευση των ανιχνευτών μας και στην ανάπτυξη της μεθόδου που χρησιμοποιεί ο αλγόριθμος για τον εντοπισμό πιθανών απειλών.

Για να αξιολογήσουμε τον προτεινόμενο αλγόριθμο αρνητικής επιλογής για την ανίχνευση κινδύνου σύγκρουσης, πραγματοποιήσαμε μια ενδελεχή ανάλυση ευαισθησίας των βασικών παραμέτρων του. Η ανάλυση αυτή εξυπηρετούσε διπλό σκοπό: πρώτον, μας επέτρεψε να κατανοήσουμε βαθύτερα τον τρόπο με τον οποίο οι εν λόγω παράμετροι επηρεάζουν τόσο την ακρίβεια όσο και τον χρόνο εκτέλεσης του αλγορίθμου. Δεύτερον, μέσω αυτής της ανάλυσης μπορέσαμε να βελτιστοποιήσουμε αυτές τις παραμέτρους, επιτρέποντάς μας τελικά να επιλέξουμε τις βέλτιστες τιμές για κάθε παράμετρο και να βελτιώσουμε τη συνολική απόδοση του αλγορίθμου.

Αφού ολοκληρώσαμε την ανάπτυξη του αλγορίθμου μας, αξιολογήσαμε την απόδοσή του όσον αφορά την ακρίβεια και τον χρόνο εκτέλεσης μέσω μιας σειράς πειραμάτων. Επιπλέον, μελετήσαμε την πιθανή πρακτική εφαρμογή του αλγορίθμου μέσω της μελέτης κάποιων πιθανών σεναρίων (case studies).

Ο προτεινόμενος αλγόριθμος φαίνεται να παρουσιάζει ελπιδοφόρα αποτελέσματα. Ο χαμηλός χρόνος εκτέλεσης, ακόμη και με περιορισμένους υπολογιστικούς πόρους, και η υψηλή ακρίβειά του καταδεικνύουν τις δυνατότητες των αλγορίθμων αρνητικής επιλογής στην αντιμετώπιση του προβλήματος εντοπισμού κινδύνου σύγκρουσης. Αδιαμφισβήτητα

υπάρχουν περιθώρια βελτιστοποίησης και περαιτέρω προόδου, πιστεύουμε ωστόσο ότι ο αλγόριθμός μας μπορεί να χρησιμεύσει ως βάση για μελλοντικές μελέτες σε αυτόν τον αναδυόμενο τομέα.

Abstract

Maritime transportation safety is threatened by the occurrence of ship collisions which may result in catastrophic consequences. Researchers have developed collision prevention methods that can be broadly classified into collision risk detection and collision avoidance. Collision risk detection aims to identify potential collisions prior to their occurrence, whereas collision avoidance aims to determine evasive actions after a potential collision has been detected. This thesis focuses on the development of a novel collision risk detection method utilizing a negative selection algorithm.

In order to achieve our goal of developing a novel collision risk detection method, we drew upon knowledge from both the ship collision risk detection and artificial immune system sectors. The initial stages of our thesis focused on reviewing and synthesizing existing research in these fields, identifying key insights that inspired our approach. From there, we developed a secondary algorithm to construct the dataset needed for training and evaluating the negative selection algorithm. This simulation algorithm incorporated an expert-based method of ship domain as a collision criterion. With the dataset in place, we then proceeded to train our detectors and develop the method used by the algorithm to identify potential threats once a detector set has been acquired.

To evaluate the proposed negative selection algorithm for collision risk detection, we conducted a thorough sensitivity analysis of its key parameters. This analysis served a dual purpose: firstly, it allowed us to gain a deeper understanding of how these parameters affect both the accuracy and the execution time of the algorithm. Secondly, through this analysis we were able to optimize these parameters, ultimately enabling us to select the optimal values for each parameter and improve the overall performance of the algorithm.

After finalizing the development of our algorithm, we evaluated its performance in terms of accuracy and execution time through a series of experiments. Additionally, we conducted several case studies to demonstrate the practical application of our algorithm.

The outcomes of our proposed algorithm exhibit promising results. The low execution time, even with limited computational resources and its high accuracy demonstrate the potential of negative selection algorithms in addressing the collision risk detection problem. Although there is room for optimization and further advancement, we believe that our algorithm can serve as a foundation for future studies in this emerging field.

Contents

List of Figures

List of Tables

Introduction

Maritime transportation is a crucial component of global trade and commerce, responsible for transporting the vast majority of international cargo. However, it also poses a significant risk to the environment and human life, particularly in the case of ship collisions. Collisions between vessels can cause severe environmental damage, as well as injuries and loss of life for crew members and passengers. Additionally, collisions can result in significant economic losses due to the damage to the vessels involved, as well as the potential for cargo loss or delay. Developing effective and reliable methods for detecting ship collisions is therefore essential for ensuring the safety and sustainability of maritime operations.

In 1972, the International Maritime Organization (IMO) made its initial efforts to develop a single set of international rules and practices aimed at preventing ship collisions. This led to the adoption of the International Regulations for Preventing Collisions at Sea (COLREGs), which were implemented on 15 July 1977 [1]. Prior to their adoption, individual countries had their own distinct sets of regulations. Currently, 155 states have adopted the COLREGs, accounting for 98.7% of the world's merchant fleet, as of 2013 [2]. The COLREGs are comprised of 41 rules divided into six sections: Part A - General; Part B - Steering and Sailing; Part C - Lights and Shapes; Part D - Sound and Light signals; Part E - Exemptions; and Part F - Verification of compliance with the provisions of the Convention. The COLREGs serve as a vital guide for the safe navigation of vessels, in conjunction with the implementation of the Automatic Identification System (AIS) and the establishment of traffic separation schemes in narrow waterways. These measures have led to a significant reduction in the frequency of ship collisions. However, it is important to note that the application of the COLREGs is ultimately carried out by seafarers, and the risk of collisions remains, as the human factor plays a critical role in ensuring safe navigation at sea.

In a recent survey by Allianz Group [3], it is stated that between 75% and 96% of ship accidents can involve human error. In another Allianz report [4] after the analysis of 15000 liability insurance claims, it was calculated that in 75% of those claims human error was the primary factor. That is equivalent to 1.6 billion dollars in losses. Seamen involved in collisions have stated that the main cause of human error is fatigue and stress. It is a fact that after the economic crisis of 2008 there is a trend of reducing the number of crew members which leads to extended fatigue and stress [7]. The need for a system that can assist the crew members in collision avoidance is unquestionable and marine researchers are working toward this direction.

Researchers initially focused on creating navigational assistance systems to improve the situational awareness of human operators in order to avoid collisions. Recently, there has been a significant increase in interest in autonomous vehicles, which use machine learning algorithms in order to avoid collisions. Therefore, two main approaches for avoiding collisions appear in research: conflict detection and conflict resolution [5]. Conflict detection involves identifying potential collisions before they happen, while conflict resolution involves finding a way to avoid or mitigate a collision once it has been detected. Both approaches aim to prevent collisions, but they focus on different aspects of the problem. According to Huang et al [5] the research in the last years focuses on two perspectives, assisting human on board and eliminating human factors. Enhancing the situation awareness of Officers on Watch (OOW) on board has been researched continuously since the 1950s [8], some of the techniques produced from this effort were the ship domain and automatic radar plotting aid. On the other hand, the development of autonomous systems that can automatically find collision-free solutions promises to replace the human factor and consequently eliminate human error. Although the focuses of these two kinds of studies may differ, most scientists believe that the research for the development of unmanned vehicles can benefit the studies of manned ships and vice versa [9].

The approach used in this thesis belongs to the research field of Artificial Immune Systems. Artificial immune systems (AIS) are computational models inspired by the natural immune system. These systems have been developed in the past few decades as a response to the need for novel and effective techniques for solving complex problems. The natural immune system is a complex biological system that can recognize and eliminate invading pathogens. The artificial immune system attempts to emulate the natural immune system by using a set of algorithms that simulate the immune response.

One important branch of AIS research is the negative selection algorithm (NSA). NSA is a type of anomaly detection algorithm that uses the concept of self and non-self to identify anomalous patterns in data. In NSA, the set of self-patterns is represented by a set of randomly generated detectors, while the set of non-self patterns is represented by the data that needs to be analyzed. The NSA algorithm compares the data with the set of detectors and flags any patterns that do not match the self-patterns as anomalous.

The development of NSA has been a subject of active research in recent years. Many variants of the original NSA algorithm have been proposed, each with its own strengths and weaknesses. For example, some researchers have proposed using hybrid approaches that combine NSA with other machine learning algorithms to improve performance. Others have focused on improving the efficiency and scalability of NSA to enable its application to larger and more complex datasets.

In summary, the development of artificial immune systems, and in particular the negative selection algorithm, has been a subject of active research in recent years. These algorithms have the potential to solve complex problems in a variety of domains, including maritime transportation. The use of NSA for ship collision risk detection is a promising area of research that could significantly improve the safety of maritime transportation.

The primary objective of this thesis is to devise an algorithm that can accurately assess the occurrence of a collision between two ships based on their respective positions, velocities, and headings. The proposed algorithm is categorized under research on collision risk detection and will ultimately output a conclusion regarding the existence of a potential collision risk.

In this endeavor, we aimed to integrate knowledge from both the domains of Artificial Immune Systems (AIS) and collision risk detection to develop our algorithm. Subsequently, we formulated a secondary simulation algorithm to generate the dataset essential for training the detectors and evaluating the algorithm. Furthermore, we conducted a sensitivity analysis to examine the primary parameters of the system, which facilitated an understanding of the algorithm's operation and provided optimal values for these parameters. Following the development of the algorithm, we evaluated its accuracy and execution time. Finally, we presented case studies to provide a comprehensive understanding of the algorithm's functioning. The contents of the present thesis can be summarized as follows.

Chapter 1 introduces the problem of ship collision risk detection and outlines the motivation and objectives of the thesis. The chapter provides an overview of the previous work done in this field and identifies the gaps in the existing literature that the thesis aims to address.

Chapter 2 presents the theoretical background and related concepts required to understand the methodology used in this thesis. The chapter covers the principles of artificial immune systems (AIS), specifically focusing on the negative selection algorithm. The algorithm developed in this study is a negative selection algorithm, making it a part of the broader framework of AIS.

Chapter 3 describes the methodology used to develop the collision risk detection algorithm. The chapter explains how the dataset used for training and evaluation was generated through a secondary simulation algorithm and provides a detailed description of the main algorithm's design and implementation.

Chapter 4 presents the results of the sensitivity analysis conducted on the main parameters of the algorithm. The chapter discusses the impact of different values for these parameters on the performance of the algorithm and provides recommendations for the optimal values to be used.

Chapter 5 assesses the accuracy and execution time of the collision risk detection algorithm and provides a concise analysis of the findings.

Chapter 6 presents some case studies that showcase the practicality and the accuracy of the proposed algorithm in real-life-like scenarios.

Chapter 7 concludes the thesis by summarizing the main contributions of the research, discussing the limitations of the study, and suggesting directions for future research in the field of ship collision risk detection.

1.1) Collision avoidance

1.1.1) First steps in collision avoidance

The increase of shipborne commercial trading and the advancement of radar after World War 2 brought the development of qualitative studies on marine collision maneuvers [8]. These early studies primarily focused on interpreting collision regulations and discussing their effectiveness and limitations, particularly in close range encounters. The main areas of focus for these studies were the behavior of marine traffic as a whole and the development of optimal strategies for evasive maneuvers in close range encounters. In the 1960s, voyage planning and the use of shore-based navigation advising systems for weather information became more widespread. During this time, collision avoidance maneuvers were typically based on radar data and the officer on watch (OOW) would make decisions on how to respond based on their interpretation of the situation. However, this approach could be unpredictable, particularly in situations involving multiple ships. During this time, it was not possible to rely on instrument-based collision avoidance maneuvers because sophisticated radar systems were not readily available. Nevertheless the ever growing shipborne trade and the lack of more sophisticated rules lead to a significant rise of ship collisions. A 1963 report from the Liverpool Underwriters Association [2] found that 21 collisions had resulted in total ship loss, a sharp increase from a five-year average of 13.8. The primary cause of these accidents was deemed to be human error when the ships were operating at high speed in congested waters.

To address these challenges, the Institutes of Navigation in West Germany, France, and the United Kingdom proposed several ideas [2] to improve safety in congested areas, including the implementation of one-way traffic schemes similar to those used on land. These ideas were well received by the International Maritime Organization (IMO) and as a result, a traffic separation scheme was implemented in the Dover Strait in June 1967 [2]. This resulted in a significant reduction in the number of collisions between ships on opposing headings. Similar schemes were later implemented by other governments around the world, particularly in areas with heavy traffic, such as the Malacca Strait.

While traffic management schemes were effective in reducing the number of collisions, they only partially solved the problem as crossing traffic still posed a threat to the regulated traffic flow. In 1975, Philips [10] proposed a conceptual marine traffic system based on land traffic regulations, but this idea was not practical because it is not feasible to install traffic lights at sea.

Subsequent research on collision avoidance focused on the development of models that integrated expert knowledge with motion prediction methods for ships. These models aimed to improve the accuracy of collision avoidance predictions as well as trying to minimize the individual human factor in decision making. Below, we will discuss some of the primary trajectory and motion prediction models that have been developed for collision avoidance.

1.1.2) Motion prediction in collision avoidance

The process of collision avoidance starts from the observation of the Target Ships (TSs) in relevance to Own Ship (OS). In the context of collision avoidance methods, the terms "target ships" and "own ships" are often used to describe the relative positions and movements of the ships involved. These terms are used from the perspective of the ship in question. For example, if Ship A is trying to avoid a collision with Ship B, Ship A is the "own ship" and Ship B is the "target ship". The majority of researchers and modern bridge systems, such as Integrated Navigation Systems (INS), agree that the second step in collision avoidance is the estimation of the motion of the ships involved in order to determine if there is a risk of conflict. This involves predicting the future positions and movements of the ships in order to assess whether a collision is likely to occur. The flow diagram of the basic steps of collision avoidance for most researchers can be seen in Fig.1, which shows the information flow of Huang's research [5] on collision avoidance.

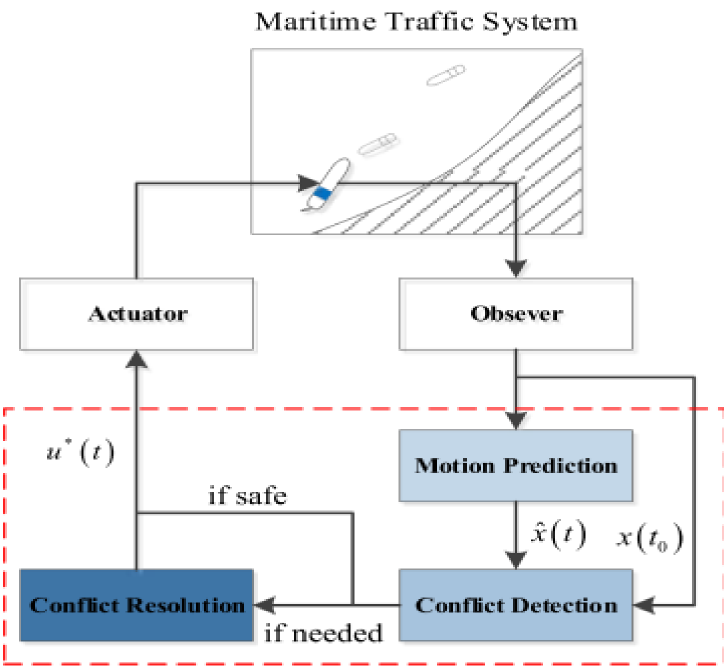

Figure 1: The information flow of collision prevention on board ships, [5]

According to Huang, ''motion models are categorized as holonomic models (constraints on configurations only) and non-holonomic models''. In a large number of studies such as Li and Jilkov [11], Benjamin et al. [12], Degre and Lefevre [13], estimating the ships' motion is based on the assumption that the ship is a holonomic vehicle that moves freely in a horizontal plane. This is the simplest way of predicting motion and it involves a constant velocity that in combination with the ship's heading produces the future positions of the ship. Nevertheless, the holonomic model does not take into account constraints on tangent C-space[1], such as the acceleration of linear and angular speed.

[1] *Tangent C-space: In motion planning and collision detection, the tangent C-space is a configuration space that encompasses the position and orientation of an object. It combines translational and rotational degrees of freedom. [14]*

To address this issue, some studies have proposed the use of kinematic motion models. These models consider the constraints on the movement of a system and can provide more accurate predictions of its future motion. Such studies are those of Hvamb [15] and Fossen [16]. Although kinematic models produce more precise predictions, they do not consider the mass of a ship, which can significantly impact its motion. These models also do not accurately account for the complex mechanisms that influence acceleration in different directions.

To improve the accuracy of prediction, researchers have introduced kinetics relations into kinematic models to take these factors into account. These models are characterized by Huang as dynamic models. Fossen in his work [17] produced a kinetic relation that contains the mass of the ship as well as the Coriolis effect and damping matrices.

Another group of dynamic models for predicting motion are Mathematical Model Groups (MMG). The MMG model is a dynamic model that is used to predict ship maneuverability. Instead of using forces as inputs, the MMG model uses rudder angle and propeller revolutions as inputs and uses empirical formulas to model the responses of hydrodynamic forces to these inputs. This approach allows for the consideration of more detailed information about the rudders and propellers, such as their specifications. As a result, the MMG model is often used in the theoretical analysis of ship maneuverability. Yasukawa and Yoshimura [18] in their research show some details of MMG. He et al [19], Li et al [20] and Xue et al [21] have applied MMG in their collision prevention techniques. While the MMG model can produce relatively accurate predictions of ship trajectory, it has a high computational cost and requires a detailed understanding of the ship's hull, rudder, and propeller. Additionally, the relationships between control inputs and forces are nonlinear and complex, which makes the MMG model less popular for the design of ship controllers and observers. As a result, researchers often prefer to use simplified models based on the MMG model to predict ship trajectory.

It is usual for researchers to avoid using overcomplicated dynamic methods in their work. Instead they prefer to simplify the aforementioned methods in terms of the dynamic response of the ship and other factors.

The motion prediction method serves as the foundation for predicting the trajectory of both the own ship and the target ship. Below we will present some of the trajectory prediction methods that have been used.

1.1.3) Trajectory prediction in collision avoidance

When dealing with predicting the trajectory of a vessel in a typical collision avoidance method, it is safe to say that dealing with the OS and the TS can be very different, since the amount of information you get from each vessel varies greatly. Regarding OS's trajectory prediction it comes down to the researcher to decide which motion prediction method they will use depending on the accuracy they want to achieve. TS's trajectory is more challenging, since a lot of the parameters that are used in the motion prediction model must be considered unknown. The most basic method for predicting the trajectory of the target ship is to assume that it will maintain its current velocity and not be affected by external factors. While this method is simple, it is not very accurate for collision avoidance purposes. A more realistic approach is to consider the uncertainties in the models, inputs, and external disturbances that can affect the motion of the target ship. This can help to improve the accuracy of the trajectory prediction. Nonetheless the predicted trajectory has a significant probability to be very different than the real. According to the knowledge of the TS, the trajectory prediction methods are categorized into three groups.

The first group is the physics-based methods. Physics-based methods for predicting the motion of a ship rely solely on the laws of physics, and do not consider the effects of control inputs or maneuvers. Some studies have ignored these factors entirely, while others have treated them as white noise. Huang [5] considers Kalman Filter as a preferred physics-based technique. Candeloro [22] in his research used KF along with holonomic models. Shah and his colleagues [23] used Kalman Filter with kinematic models ("simple car" model). While physics-based methods can accurately predict the short-term trajectory of a ship, they are not able to predict changes in trajectory that may result from changes in maneuvers [24]. This can limit their effectiveness in collision avoidance situations where the ships may be making sudden changes in course or speed.

Another group of trajectory prediction methods is maneuver-based methods. Maneuver-based methods consider the intended maneuvers of a ship, such as changes in course or speed, when predicting its trajectory. These intentions can be learned or estimated from historical traffic data or by following protocols for ship encounter situations, such as the COLREGs. Scheepens et al [25] produced an algorithm that learns the behavioral patterns of ships for a certain area through massive traffic data and then uses these patterns to support the prediction. Other researchers have produced models by integrating different methods for processing such data. Simsir et al [26] developed a model using neural network, Rong [27] used Gaussian process and Peel and Good used Hidden Markov Model [28].

Interaction-aware methods take into account the interactions between ships when predicting their trajectories. These methods may include communications between ships, such as the exchange of maneuver intentions or trajectory information [28, 29]. By allowing the ships to exchange their planned trajectories, these methods can provide more accurate predictions because the ships have a better understanding of their own dynamics and intentions. As a result, interaction-aware methods may be more accurate than other methods for predicting ship trajectories. In Fig. 2 we can see an illustration of the different prediction methods we mentioned [5].

After a method for predicting the trajectory of a ship has been chosen, it is necessary to determine whether the navigational state of the ship poses a risk of collision. If a collision risk is identified, an evasive action may be necessary. This action can be taken by the officer on watch (OOW) or by an autonomous system on board the ship that is responsible for collision avoidance. The identification of potential collisions is the scope of conflict detection. As the current thesis aims to develop a method for detecting conflicts, the next chapter will discuss the research that has been done in this field.

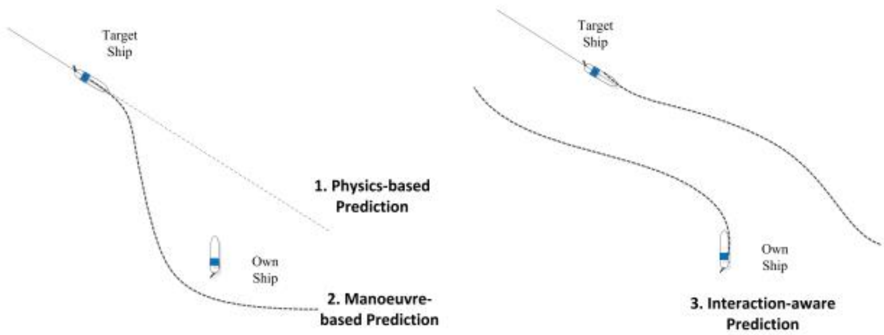

Figure 2: An illustration of different prediction methods, [5]

1.2) Conflict/Collision Risk Detection

As already mentioned conflict detection is an integral part of all collision avoidance methods. In research we find some variations in the definition of conflict detection. Goerhand [30] and Kao [31] define conflict detection as the process of identifying potential collisions and alerting the officers on board the ships and the operators in Vessel Traffic Service (VTS) of the risk. Kuwata [32] suggests that the role of conflict detection is to trigger the autonomous system on board to find evasive actions. Johansen [33] notes that conflict detection has the additional role of evaluating the risk of alternative paths or evasive actions. In his work Huang [5] categorizes the collision risk detection methods in two groups, the “expert-based methods” and the “model-based methods”.

Expert-based methods rely on the direct use of expert knowledge to assess the risk of collision. These methods produce a risk measure that reflects the beliefs of the experts about the likelihood of a collision occurring. On the other hand, model-based methods use a simplified model to describe the physical processes of collision and estimate the probability of a collision event occurring. The risk measure produced by these methods is a conditional probability of collision. Within each of these two categories, there are several different ways to represent the collision risk. Some researchers prefer to present the risk as a numerical value, while others use graphical representations such as rings of warning or action lines on a two-dimensional map. These different representations can help to communicate the risk of collision to the OOWs and VTS operators in a way that is easy to understand and respond to. Below we will present and discuss some of the expert-based methods.

1.2.1) Expert-based method

1.2.1.1) Collision risk index

One common approach to conflict detection involves the use of a numerical value called the collision risk index (CRI). Methods in this category trigger a collision alarm when the CRI exceeds a predetermined threshold. The way in which the CRI is calculated, and the value of the threshold are often based on the expertise and experiences of captains, pilots, and other maritime professionals. As a result, these methods are typically considered to be expert-based methods, as they rely on the beliefs and judgement of experts to assess the risk of collision. This method usually utilizes some indices to measure the risk. These are the closest point of approach (CPA) which refers to the shortest instantaneous distance between the ships involved [8], the time to the closest point of approach (TCPA) and the distance to the closest point of approach (DCPA). As Tam [8] states the initial concepts for assessing collision risk were based on those indices. Usually, researchers use a polynomial equation that combines DCPA and TCPA to calculate a single value, that of the CRI. A generalized form of CRI measurement is shown as follow:

$$CRI_1 = w_1 f(DCPA) + w_2 f(TCPA)$$

Where w_1 and w_2 are weights determined by utilizing the experts' knowledge. The integration of this knowledge can be achieved through various techniques. Lee and Rhee have used fuzzy theory [34]. Fuzzy theory can be seen as a great tool in these methods, since the experts' knowledge can be characterized as imprecise for mathematical standards. On the other hand, Chin and Debnath have used Probit Regression [35] in order to achieve a proper interrelation between the weights and the experts' beliefs.

In order to improve the accuracy of CRI researchers try to introduce new risk indicators that provide more detailed information about the encounter between ships. Li and Pang introduced relative distance in their calculation [36], while Zhao et al. used relative bearing [37] and Gang et al used ratio of speeds [38].

In his work Lisowski [39] introduced a non-linear equation in order to calculate CRI. This method uses a Euclidean norm that contains DCPA, TCPA and relative distance. Instead of using formulations, Ozoga and Montewka designed a pre-set matrix table in order to determine the CRI value [40]. Other researchers in their measurements abandon completely the use of T/DCPA. In their work Perera and Soares used some easily obtained variables for their risk assessment. More precisely they constructed mathematical relationships around relative positions and relative speeds. They found that these measurements show the tendency of relative movement [41]. In Fig.3, we find a graphical representation of the factors used in this method [41].

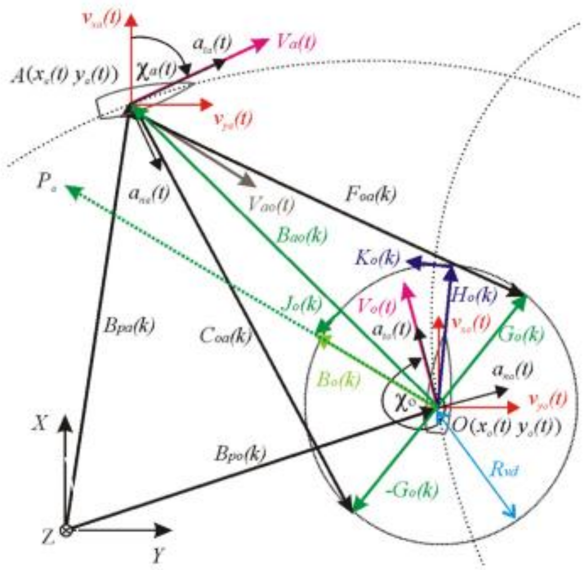

Figure 3, Perera and Soares calculation method, [41] In their research W. Zhang and his colleagues [42] used relative distance and speed along with the encounter angle in order to calculate the CRI. In this case some encounter scenarios are used to determine the parameters of the equation.

1.2.1.2) Ship Safety Domain

The concept of a ship safety domain is based on the idea that the safe distance between two ships may vary depending on the direction in which they are approaching each other [17]. This is due to factors such as the relative sizes and speeds of the ships, the characteristics of the surrounding environment, and the available maneuvering space. There are numerous papers on ship domains, in their work Szlapczynski and Szclapczynska [43] tried to sum up all the different developments in this field.

Ship safety domains can be broadly classified into three categories: those developed through theoretical analysis, those based on the expertise of maritime professionals, and those determined empirically through the analysis of data. It is common for these categories to overlap, with some models combining elements of different approaches. Empirically determined domain models are often simpler, as they are based on observed data rather than theoretical analysis. However, their potential applications may be limited to situations where the general shape and size of the domain are sufficient for statistical analysis, and more precise dimensions are not as important. Knowledge-based and analysis-based models of ship safety domains have a wider range of applications, including collision avoidance, near miss detection, and waterway risk analysis. These purposes require more detailed and complex models, which often have many parameters that take into account multiple factors that contribute to collision risk. These models are generally more flexible and can be used in a wider range of situations, but may be more complex and require more specialized expertise to develop and interpret.

All models of ship safety domains are influenced by the characteristics of the waterway in which they are applied, although the extent of this influence may vary. For example, when evaluating the capacity of a waterway or assessing the overall risk of a particular region, the shape, traffic density, and patterns of vessel traffic may be key factors. In the context of collision avoidance, the type of water way (such as narrow waterways, restricted but wider waters, or open waters) may be more important in determining the appropriate safety margins and maneuvers. In any case, the waterway plays a role in shaping the characteristics of the ship safety domain and the associated risks and challenges.

Ship domain was first introduced by Fuji and Tanaka in 1971 [44] when they studied the maritime traffic in Japanese channels. The ship domain they produced was of an ellipsoidal shape with OS at the center. Their research along with the works of Coldwell [45] and Goodwin [46] were based on statistically processed radar data. The radius of the domain according to Goodwin should also take into account many independent variables as the type of area, traffic density, ship length and maximum speed.

The definitions of ship domain given by these three authors are very similar, yet they have been interpreted in multiple ways by various authors. Fuji defined ship domain as follows "*a two-dimensional area surrounding a ship which other ships must avoid – it may be considered as the area of evasion*". Goodwin defined it as "*the effective area around a ship which a* *navigator would like to keep free with respect to other ships and stationary obstacles*". In Coldwell's work we can find ship domain as "*the effective area around a ship which a typical navigator actually keeps free with respect to other ships*". The different interpretation of these definitions has led to different safety criteria applied in the research of ship domain. In their work Szlapczynski and Szclapczynska [43] show the four safety criteria used in research as follows:

a) Own ship's (OS) domain should not be violated by a target ship (TS)
b) A target ship's (TS) domain should not be violated by the own ship (OS)
c) Neither of the ship domains should be violated (a conjunction of the first two conditions)
d) Ship domains should not overlap – their areas should remain mutually exclusive (the effective spacing will be a sum of spacing resulting from each domain)

The different safety criteria can be seen in Fig.4 [43].

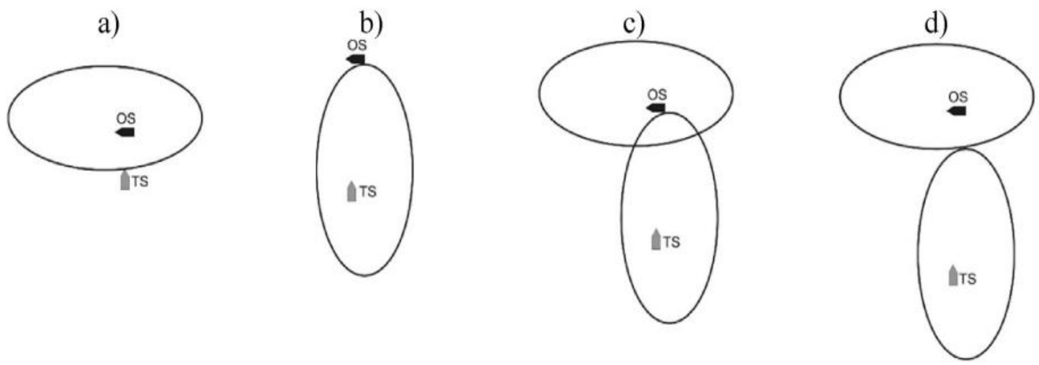

Figure 4: Different domain based criteria: a) OS domain is not violated, b) TS domain is not violated c) both TS and OS domains are not violated d) TS and OS domains do not overlap. [43]

The first two criteria give different estimations of safety even if the same domain is used since they are dependent on which ship does the assessment. For the other two if the same domain is applied, the assessment would be the same regardless of the point of view. For this reason the latter two safety criteria can be considered as symmetrical. It should be pointed out that the last criterion has been used in the late research by Rawson et al. [47] and Wang and Chin [48] and it is by far the strongest in terms of safety.

It becomes obvious that the criterion that will be chosen gives a very different minimum spacing that should be kept, even if the domain dimensions remain the same. Regarding the domain dimensions, the first papers on the subject calculated them mostly in accordance with the ships' length. One such example is that of Coldwell's ship domain. In later research we find different expressions about domain's size. In their work Wang and Chin [48] assume that the size of the domain is a linear function of ship's length and a quadratic function of its speed. The shape of the domain in this work varies since the safe distance depends on the polar angle measured clockwise from the ship heading.

In ship collision risk detection, ship domain is usually used to visualize the collision risk by a set of warning rings surrounding the OS. When a target ship enters or will enter this region, a collision alarm is used.

1.2.2) Model-based method

1.2.2.1) Binary collision criteria

Binary collision criteria are used to provide a deterministic assessment of whether a collision will occur or not based on a given scenario. One common criterion is the DCPA along with the assumptions that the target ship maintains a constant velocity and that both ships are represented as circles. This criterion offers a simple and straightforward way to evaluate the likelihood of a collision, but may not accurately reflect the complexity and variability of real-world maritime situations. Nevertheless, this method is widely used for manned and unmanned vessels [49].

1.2.2.2) Probability of collision

The occurrence of collisions in a marine system can be characterized as a stochastic process. The probability of collision can be used to quantitatively assess the risk of dangerous encounters. However, the outcome of such encounters may be influenced by various sources of uncertainty, including errors in sensor readings, the effects of environmental factors, the response of the targeted system, and inaccuracies in predictions [50]. Given the complexity of these factors, it may be infeasible to account for all sources of uncertainty, particularly when certain uncertainties are difficult to measure. An alternative approach to handling the complexity of uncertainties is to assume that certain aspects of the system are known, and only certain sources of uncertainty are taken into account. Additionally, it is commonly assumed that the probability distributions for these uncertainties are already known. With these assumptions, the probability of collision can be calculated based on the specified uncertainties.

Shah et al. [49] used Monte Carlo simulation in order to calculate the probability. Monte Carlo simulation is a method for using random sampling to perform numerical simulations and estimate a range of possible outcomes for a given system or model. It is considered particularly useful when the underlying model is complex or difficult to solve analytically. Park and Kim [51] in their research utilized a concept of probability flow for the calculation. In probability theory and stochastic analysis, the concept of a probability flow refers to a mathematical object that describes how probability evolves over time in a given system. A probability flow is often represented mathematically as a family of probability measures on a state space, where each measure represents the probability of different outcomes at a specific point in time. The probability flow is useful to model uncertain situations where the outcome or state of the system at some point in time depends on the outcome or state at some previous time.

Although researchers have found some methods of calculating the probability of a collision, they have not found a way to set a threshold for which the collision alarm would be launched.

1.2.2.3) Dangerous region

One approach to collision detection involves compiling a set of speed or course values for the own ship (OS) that are known to lead to collisions with the target ship (TS) and displaying this set of values to the officer on watch (OOW). A collision warning is activated when the current velocity of the OS falls within this set. This set can be found in research with many different names such as Collision Threat Parameter Area (CTPA) in Lenart's research [52] and Velocity Obstacle in Huang et al research [53]. Initially, to establish the set of conditions for collision threat assessment/collision detection system (CTPA/CDS), certain simplifying assumptions were made by researchers, such as the target ship (TS) maintaining a constant velocity and heading, and the assumption that both the TS and the other ship (OS) are circular in shape [52]. In more recent research, scientists have adopted different practices and concepts in order to make their models more realistic. For instance, Szlapczynski and Szlapczynska utilized ship domain [54] and Huang [53] considered changes in TS speed and course. As the set of velocities of the other ship (Velocity Obstacle) that are considered dangerous is determined within the solution space, some researchers have proposed using the proportion of safe solutions as a metric. Another approach is to directly display a region of the waterway where a collision with the target ship (TS) is likely to occur, which is typically positioned at the closest point of approach (CPA). A collision warning is issued when the current velocity of the other ship (OS) indicates that it may enter this dangerous area. One well-known method in this category is the Predicted Area of Dangers (PAD) proposed by Zhao-lin in 1988 [55], or the Projected Obstacle Area (POA) proposed by Gerhart et al. in 2006 [56]. Other methods that have been proposed include the Obstacle Zone by Target (OZT), whose size and position are determined using a joint probability, as proposed by Fukuto and Imazu in 2013 [57], and Kayano and Kumagai in 2017 [58]. Another example is the Fuzzy Collision Danger Domain (FCDD) which takes into account multiple factors to determine its size, as proposed by Su et al. in 2012 [59].

1.2.2.4) Action lines

This group of studies try to construct an action line that surrounds the OS in its geographical space. This line is used to show the last chance the OS has to evade a collision via a specific evasive action. The concept is very similar to, and it is a form of a ship domain. In contrast to the ship domain, the determination of the action is based on simulation and not on the experts' beliefs. The main assumptions that the researchers take in order to produce the action line are the following. First, the TS keeps its initial speed and heading. Second, the OS takes a specific evasive action, e.g., a hard-port turn. Through multiple simulations, a set of initial positions of the TS that the OS can avoid collision with via the fixed action is found. Sczlapczynski et al depicts these positions as an action line [60]. Krata and Montewka through these simulations construct the concept of critical distance which is very similar to the action line [61]. Baldauf et al. [62] presented a similar concept, with the line named as the last line of defense. By performing simulations for different fixed evasive actions, different action lines are produced, that can be used as collision alarms.

1.2.3) Conclusion

It is certain that obtaining an accurate collision risk at each time step is extremely difficult if not impossible. This comes down to the great number of uncertainties that relate to a ship encounter, from the weather conditions to the intentions of the OOWs in the TS. Nevertheless, neither man nor machines can achieve collision prevention without collision risk detection. As we show above researchers have developed methods that try to provide reliable solutions to the risk detection problem. One group is trying to integrate the knowledge of the experts in methods that can assist the OWWs and VTS operators by improving their situational awareness. Although these methods can be of great help as a risk- informed tool, they can be biased and disregard the physical process of the collision. The other group of methods tries to employ technical knowledge regarding the ship movement and the physics of a collision in order to define the probability of a collision. In order to perform such calculations, the researchers need to minimize the related uncertainties. To achieve that they study ideal cases. The ideal case can be the worst case, normal case, or the most possible case. This type of method can provide the users with feedback that is easy to use, a prime example is that of DCPA. On the other hand, it becomes obvious that an ideal simplified scenario cannot cover the complete environment that the OS faces.

Another diversification of the existing methods for collision risk detection is the representation of the collision risk. One way is to use a numerical value to represent the risk. Another way is to graphically depict the risk, which is the approach followed by Szlapczynski and Szlapczynska [43]. Another way is to present the risk as acceptable or unacceptable in a binary way.

In their work, Huang et al. [5] conclude that expert-based methods are more capable of meeting different users' perception of risk in a given scenario. Therefore, these methods are more widely used in studies for supporting manned ships. The main drawback about these methods according to the authors is that they cannot give an explicit judgment to users about whether the collision will occur. For this reason, the model-based method find use these methods may lack the preciseness of expert-based methods, but they can provide a relative objective answer, based on which the boundaries of safety and danger are made clear.

Artificial Immune Systems

2.1) Introduction

The term “Artificial Immune Systems” (AIS) is used to describe a class of computationally intelligent, rule-based machine learning systems inspired by the principles and processes of the vertebrate immune system [63]. All the different AIS techniques that have been developed try to use and mimic different mechanisms of the natural immune system in order to solve real world science and engineering problems such as machine learning, pattern recognition, data mining, intrusion detection and engineering system optimization etc [63]. The concept of AIS was introduced in the mid-1980s with the work of Farmer, Packard and Perelson [72]. Main examples of AIS techniques are the Artificial Immune Networks, the Clonal Selection Algorithm, and the Negative Selection Algorithm.

Artificial Immune Networks are inspired by the idiotypic network theory proposed by Jerne [65] that describes the regulation of the immune system by an idiotypic network of interconnected B-cells. This type of algorithm is used in clustering, data visualization and analysis [64].

Clonal Selection Algorithms were introduced by De Castro and Von Zuben [73] for learning and optimization. These algorithms focus on the Darwinian inspired theory by Burnet of clonal selection of acquired immunity, which state that only the immune cells that recognize the antigens can proliferate and produce clones subject to mutation.

Negative Selection Algorithms were proposed for the first time by Forrest et al [76] to detect data manipulation caused by a virus in a computer system, and more generally for anomaly detection. They are inspired by the mechanism of the immune system that involves training T-Cells to recognize antigens and to prevent them from recognizing the body’s own cells. Negative selection refer to the destruction of the self-reacting cells.

2.2) Immune System-Basic Components

Immunology is an extended topic of the field of medicine. Therefore, some basic features that are deemed relevant to the concept of this thesis will be promptly presented below.

The immune system is a highly intricate network comprising of organs, cells, and molecules that work in tandem to combat invading pathogens. The cellular components of the immune system originate in the bone marrow and migrate to peripheral tissues, circulating in the blood and lymphatic system. The primary function of these cells is to discern between self and non-self cells, with the latter requiring further identification to determine the appropriate defense mechanism. A remarkable aspect of the immune system is its ability to evolve and differentiate between foreign antigens and host cells, and its multilayered structure that forms an intricate defense mechanism against infection. Various layers of the immune system provide a range of barriers against infection, as illustrated in Figure 5 [63].

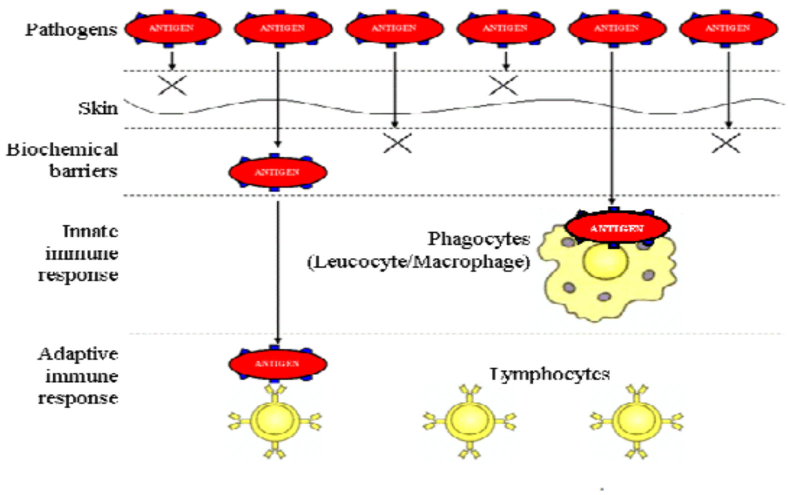

Figure 5: Multiple layers of immune system, [63]

Following penetration through the initial physical and biochemical barriers, including the skin, saliva, tears, mucous secretions, and pH concentration, antigens are encountered by the innate and adaptive immune system to prevent contamination of the organism. The innate system, comprised of a diverse array of leukocytes and macrophages, serves as the first line of defense to counteract a portion of the invading antigens. In contrast, the adaptive system is the most sophisticated defense mechanism of the organism, developing over time to recognize and eliminate diverse antigens through specialized white blood cells called lymphocytes. Two primary types of lymphocytes, B- and T-cells, are generated in the bone marrow, with T-cells undergoing further maturation in the thymus. Prior to maturation, both cell types undergo exposure to self-proteins. Upon binding to self-proteins, the cells are eliminated, while the generation of specific antigen receptors on their surface occurs if no such binding occurs. B- and T-cells are then circulated throughout the body via the blood stream and the lymphatic system to recognize and eliminate antigens. These cells differ in their modes of operation, with B-cells responsible for the production of antibodies, which recognize a single type of antigen. The attachment of a B-cell to an antigen triggers the immune response, either by the B-cell itself or in combination with a T-cell. Additionally, the attachment of a B-cell to an antigen triggers the production of memory cells, which offer future protection in case of re-exposure to the same antigen [66].

2.3) Main Immunological Principles in Artificial Immune Systems

In this section, we will outline the immunological principles that serve as the foundation of artificial immune systems (AIS). The three primary principles that have inspired various AIS methods include the clonal selection principle, immune network theory, and negative selection mechanism. The development of AIS techniques has been heavily influenced by these principles. This thesis focuses on a method based on the negative selection principle. However, researchers are working to incorporate multiple principles into their algorithms. In the following paragraphs, we will discuss the fundamental concepts underlying these principles.

The concept of "self" and "non-self" is highly significant in the field of immunology, and it is equally essential in the context of Artificial Immune Systems. A succinct definition of these terms can be provided as follows: "Self" pertains to the cells, molecules, or components inherent to an organism that are duly recognized as belonging to the individual. On the other hand, "non-self" refers to external substances, cells, or molecules that are perceived as potentially detrimental or originating from outside the organism. The immune system is responsible for discerning between self and non-self entities, enabling appropriate responses such as self-tolerance and immune reactions against non-self agents. This discrimination mechanism is vital for maintaining the overall integrity and proper functioning of the organism.

2.3.1) Clonal selection principle

As previously mentioned, antibodies are capable of recognizing specific types of antigens. The clonal selection principle is a process in the immune system that involves the creation of antibody clones capable of defending the organism against the specific antigen at hand. When a foreign organism invades the host, it often multiplies and damages the host's cells. To combat this, the immune system replicates the cells that are successful in recognizing and fighting the invading antigen. During this cloning process, the immune cells undergo mutation that enables them to better target and eliminate the antigen. In Fig.6 we can see the basic steps of clonal selection [63].

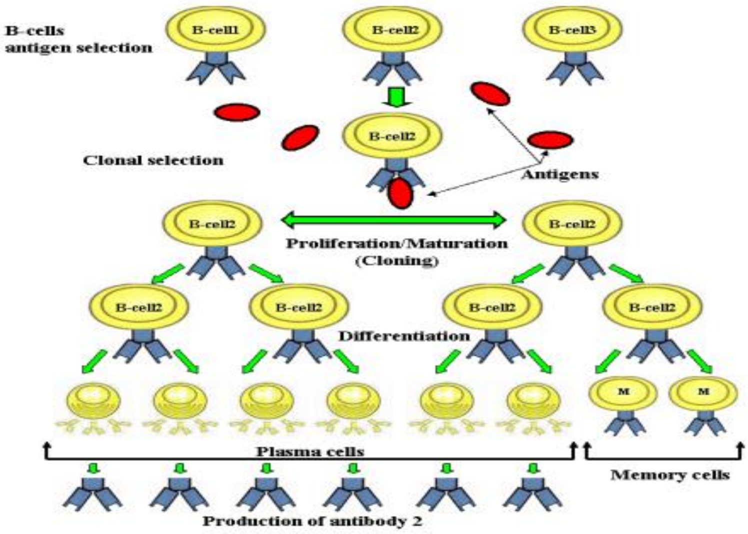

Figure 6: The clonal selection, [63]

2.3.2) Immune Network Theory

Jerne was the first to introduce immune networks as a means of explaining how the immune system maintains memory of an antigen. He suggested that B-cells form a network through which the immune system's memory is passed down [65]. Each antibody contains an "idiotope" or a "key" which endows the antibody with two capabilities. Firstly, it allows the antibody to recognize a specific antigen. Secondly, it enables the antibody to stimulate another antibody. This means that an antibody can act as an antigen to another antibody, thereby stimulating the creation of antibodies of the same type. Through the interaction of a large number of antibodies, the immune system self-regulates and self-organizes its cells by retaining and multiplying useful antibodies while eliminating useless ones. A depiction of the main components of immune network theory is given in Fig. 7 [63].

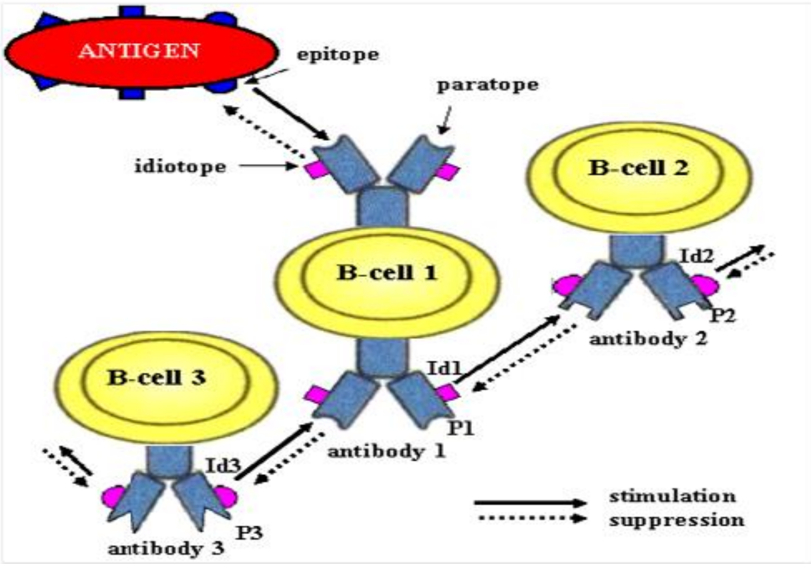

Figure 7: Immune network interactions, [63]

2.3.3) Negative Selection Mechanism

Negative selection is a crucial process for maintaining immune tolerance to self-antigens. It involves the selection and removal of immature lymphocytes that react with self cells, thereby preventing self-reactive immune responses. This process occurs through the interaction of lymphocytes with self-antigens, which triggers cell death of self-reactive lymphocytes. The goal of negative selection is to enable the immune system to recognize and respond to foreign antigens while avoiding harmful responses against self. In Fig. 8 we find a depiction of negative selection [63].

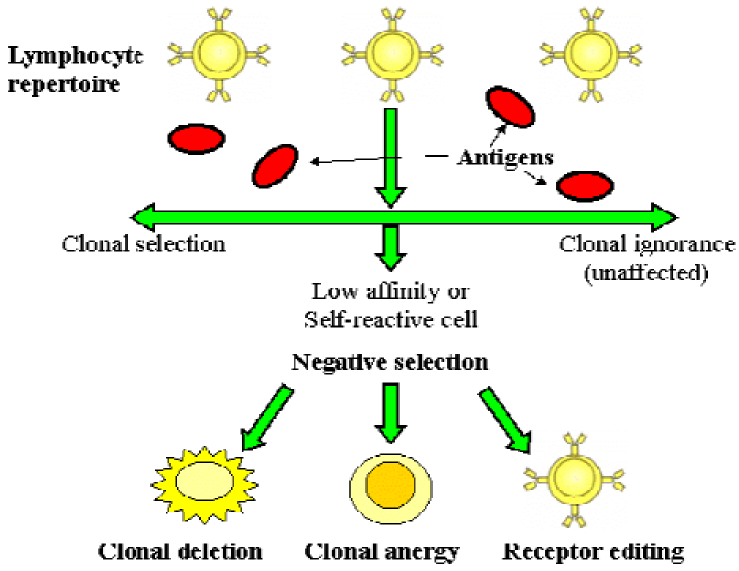

Figure 8: Negative selection progress, [63]

2.4) Artificial Immune Algorithms

Each of these main immunological principles are the inspiration behind the main types of artificial immune algorithms. In this section we will show some basic characteristics of the clonal selection algorithm, the immune network algorithm, and describe the negative selection algorithm more thoroughly.

2.4.1) Clonal selection algorithm

The Clonal selection algorithm (CSA) was first proposed by de Castro and Von Zuben in 1999 [73]. Following the clonal selection principle of the biological immune system, CSA aims to develop solutions that are of close affinity to specific “antigens”. De Castro and Von Zuben showed that CSA has a great potential of solving machine-learning tasks, such as pattern recognition and optimization. In 2002, de Castro [74] produced CLONALG which is considered by many the most characteristic clonal selection algorithm. CLONALG is considered optimal for pattern recognition although it has the potential negative aspect that when a candidate memory cell is selected, the remaining mutated clones are discarded, thus there is a possibility of discarding some qualified candidates. Since it was proposed, CLONALG has been used in many different applications ranging from voltage stability evaluation [70] to medical research on people with Parkinson’s disease [71].

2.4.2) Immune network algorithm

The Immune Network Algorithm (INA) was originally proposed by Farmer et al. [72] in 1986 and is based on Jerne's network theory. The principle underlying this algorithm is as follows: "stimulation" refers to the recognition of an antigen or another cell by the immune cell, while "suppression" refers to the recognition of the immune cell by another cell. The sum of the stimulation (Nst) and suppression (-Nsup) received by the network cells, plus the stimulation of antigen recognition (As), corresponds to the stimulation level S, which defines the reproduction and generic variation of an immune cell.

One notable application of the immune network for data analysis was proposed by de Castro and von Zuben [73], who introduced aiNET. The first adaptation of aiNET was carried out by de Castro and Timmis [74], who developed opt-aiNET (the optimization version), designed to solve multimodal function optimization problems. This evolutionary-like algorithm has several interesting features. Campelo et al. [75] later modified opt-aiNET for use in electromagnetic design optimization, creating the new version m-aiNET. The objective of m-aiNET is to optimize the computational effort required for a single run of the algorithm. By utilizing some constraint-handling techniques, the algorithm is more suitable for constrained problems. Overall, algorithms based on INA have been developed to address several complex optimization problems.

2.5) Negative Selection Algorithm

The negative selection algorithm (NSA), initially proposed by Forrest et al. in 1994 [76], is based on the mechanism of the immune system that trains T-cells to recognize only non-self antigens and not self-cells. Therefore, the main application of NS algorithms has been focused on anomaly and fault detection [81]. The NSA can be summarized as follows according to Y.M Chen and M.L. Lee [77]:

- Define self as a collection S of elements in a feature space U, a collection that needs to be monitored. In most cases U corresponds to the space of states of a system represented by a list of features, S can represent the subset of states that are considered as normal for the system.
- Generate a set F of detectors, each of which fails to match any string in S. An approach that mimics the immune system generates random detectors and discards those that match any element in the self set. However, further developments have been made in order to minimize the number of generated detectors while maximizing the coverage of the nonself space.
- Monitor S for changes by continually matching the detectors in F against S. If any detector matches, then a change is known to have occurred, as the detectors are designed not to match any representative samples of S.

In Figure 9 we can find a depiction of the feature space and how the self and non-self states exist in it [78].

The above description provides a broad overview of negative selection algorithm (NSA), but it does not provide any details about the representation of the problem space and the type of matching rule used. As per Dasgupta et al [78], creating effective detectors can be a complex process that depends on various factors, such as the type of problem space (continuous, discrete, mixed, etc.), the representation of the detectors, and the matching rule used to determine if a detector matches an element or not. It is important to consider these factors in order to design a robust and effective NSA algorithm for anomaly detection or fault detection applications.

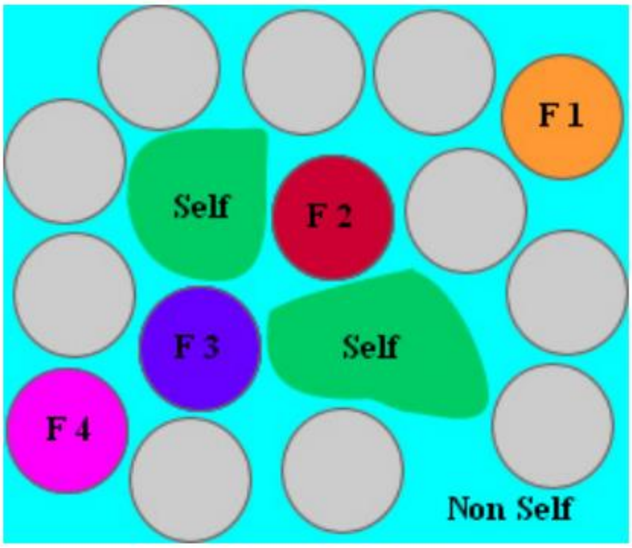

Figure 9: The concept of self and nonself in a feature space. Here F1, F2, etc. indicate different fault conditions detected by detectors, [78]

Different variations of NSA have been developed through the years in order to better address different problems. In his work Gonzalez [66] has studied the main variations of negative selection algorithms. Their main attributes are presented below.

- Negative selection with detector rules (NSDR)

This algorithm uses a genetic algorithm to evolve detectors with a hyper-rectangular shape in order to cover the non-self space. These detectors can be interpreted as If-Then rules, which try to precisely characterize the self/non-self space.

- Negative selection with fuzzy detector rules (NFSDR)

This algorithm is an extension of the NSDR algorithm in order to facilitate fuzzy rules. Fuzzy logic refers to cases where the truth of an expression is not binary (True-False). According to Gonzalez, “this produces a measure of deviation from the normal that does not need a discrete division of the non-self space.”.

- Real-valued negative selection (RNS)

This algorithm starts with a set of hyper-spherical detectors which are randomly distributed in the feature space. The algorithm changes the position of the detectors based on two main goals. The first is the maximum coverage of the non-self subspace. The second is to minimize the coverage of the self samples.

- Randomized Real-valued negative selection (RRNS)

This type is a variation of RNS algorithms. Based on a type of randomized algorithms called Monte Carlo methods, this method solves some of the drawbacks of a typical RNS algorithm. Specifically, it can produce a good estimate of the optimal number of the hyper-spherical detectors needed to cover the non-self space, and the maximization of the non-self coverage is done through an optimization algorithm with proved convergence properties.

The algorithm developed in the current thesis is mainly based on the techniques used in the real-valued negative selection algorithms and it is focused on the general concept of anomaly detection. In the next paragraph we will try to define the problem of anomaly detection.

2.6) Anomaly detection problem definition

Anomaly detection is the process which aims to distinguish normal states of a system from anomalous ones. To achieve this, it is crucial to identify the variables that accurately represent the various states of the system. Based on this principle, the following definitions can be established.

Definition 1: System state space

A state of the system is represented by a vector of *n* features, $x^i = (x_1^i, x_2^i, \dots, x_n^i) \epsilon [0,1]^n$. Each state is represented by a set $U \subseteq [0,1]^n$. It includes the feature vectors corresponding to all possible states of the system.

These features can represent current and past values of system variables. The actual values of the variables could be scaled-normalized to fit the range [0,1].

Definition 2: Normal subspace (crisp characterization)

A set of feature vectors, Self ⊆ U, represents the normal states of the system. Its complement is called Non-Self and is defined as Non-Self = U – Self. In some cases, we will define the Self (or Non-Self) set using its characteristic function: $x_{self} : [0,1]^n \rightarrow \{0,1\}$.

$$x_{self}(x) = \begin{cases} 1 \ if \ x \in Self \\ 0 \ if \ x \in Non\_Self \end{cases}$$

The terms self and non-self are motivated by the natural immune system. In most cases though there is no sharp distinction between normal and abnormal states; instead, there is a degree of normalcy (or conversely, abnormality). The following definition reflects these fuzzy cases.

Definition 3: Normal subspace (non-crisp characterization)

The characteristic function of the normal (or abnormal) subspace is extended to take any value within the interval [0,1]. The value 1 represents completely normal state and value 0 completely abnormal. Thus, any value in the interval represents a state with some degree of abnormality.

This non-crisp characterization allows a more flexible distinction between normalcy and abnormality. However, in a real system it may be necessary to decide when to raise an alarm or not. In this case, the problem becomes again a binary decision problem. By establishing a threshold value, the transition back to the crisp characterization is easy.

$$\mu_{self,t}(x) = \begin{cases} 1 \text{ } if \text{ } \mu_{self,t}(x) > t \\ 0 \text{ } if \text{ } \mu_{self,t}(x) \leq t \end{cases}$$

2.7) Anomaly detection using Negative Selection Algorithms

Negative Selection Algorithms (NSAs) have been primarily designed for tackling anomaly detection problems, which involve monitoring the state of a system. The objective is to identify abnormal states by obtaining a set of normal samples and training a sufficient number of "antibodies" or detectors to cover the abnormal states of the system. In this chapter, we will delve into the approaches used by NS algorithm developers to generate effective detectors.

2.7.1) Binary matching rules

The initial studies on NS employed binary matching rules to create antibodies-detectors that match the antigens. The r-continuous matching method, which was the first NS algorithm, used binary matching rules, as did r-chunk matching, which was proposed by Bathrop et al. [77], and Hamming distance matching rules, proposed by Farmer et al. [72]. However, despite setting the foundation for the development of NS, binary matching rules were deemed unsuitable for capturing the structure of even basic problem spaces by researchers [68], as they have a low-level representation.

2.7.2) Negative selection with detector rules (NSDR)

To effectively apply NS algorithms for knowledge extraction, a higher level of representation for the detectors is crucial. In 2002, Dasgupta and Gonzalez [80] proposed a novel approach that incorporated a more sophisticated detector generation algorithm and different matching rules, which facilitated the use of hyper-rectangles as detectors. These hyper-rectangles were contained within the subset of R^n, the unitary hypercube $[0,1]^n$, and could be interpreted as anomaly detection rules. This added structure to the detectors and enabled the extraction of high-level knowledge in the form of If-Then rules through the use of an evolutionary NS algorithm. In contrast to the early binary matching rules, which were limited by their low-level representation, this new approach allowed for a more nuanced understanding of complex problem spaces.

2.7.3) Real-valued negative selection (RNS)

In 2002, Gonzalez proposed a hybrid approach [81] for anomaly detection using a real-valued negative selection algorithm (RNS). This approach operates in a self/non-self space that is represented as a subset of R^n. A detector is defined by an n-dimensional vector that corresponds to the center and a real value that represents its radius. Therefore, a detector can be seen as a hypersphere in R^n. The detector matching rule is expressed by a function of the Euclidean distance between the detector and antigen, as well as the radius of the detector.

The RNS algorithm begins with a set of self samples represented by n-dimensional vectors, which is used to evolve another set of points (detectors) that cover the non-self space. According to Dasgupta [78], the RNS detector generation starts with a population of candidate detectors that are then matured through an iterative process. The center of a detector is chosen randomly, and the radius is a variable parameter that determines the size of the detector. During the training of the detector set, the position of the detectors is updated at each iteration based on two factors: moving the candidate detector away from self points and keeping the detectors separated to maximize the covering of the non-self space.

In Figures 10 and 11 we can see schematic representations (flow diagrams) for Dasgupta's and Gonzalez's algorithms respectively [78, 81].

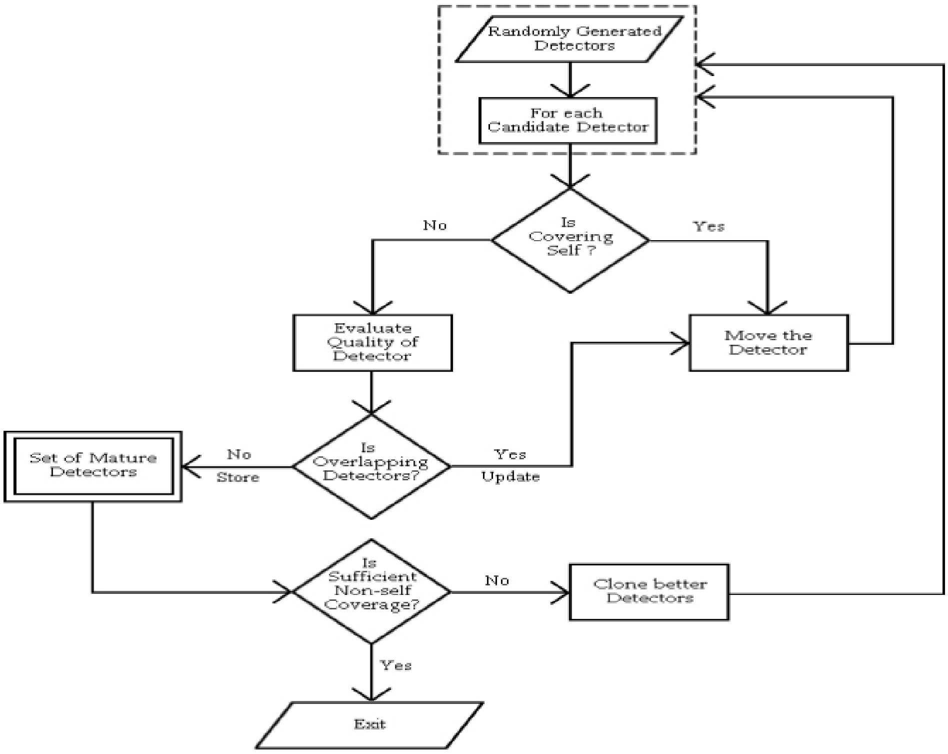

Figure 10: Flow diagram for Dasgupta's real-valued negative selection algorithm, [78]

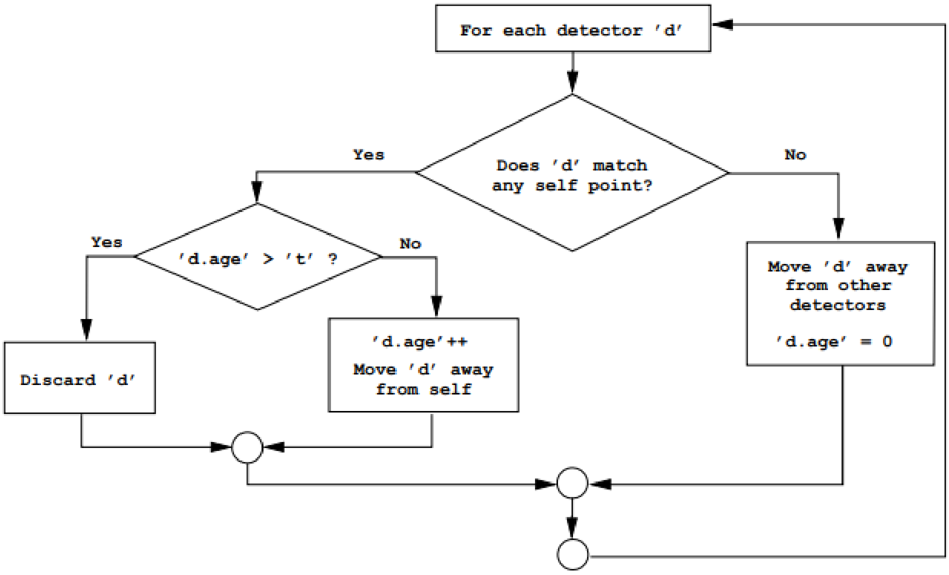

Figure 11: An iteration of Gonzalez's real-valued negative selection algorithm, [81]

The parameter r specifies the radius of each detector. For an antigen to be detected by a detector, the distance between them must be at most r. Since a detector should not match a self point, the shortest allowable distance for a detector d to the self-set is r. In Gonzalez's first development of an RNS algorithm, the candidate detectors were chosen to have a fixed radius. In this case, the radius also specifies the allowed variability in the self space. In Dasgupta's model, the radius of each detector is different and can be calculated. The radius of each detector is calculated as the Minkowski distance of the centers of the detector and its closest self point, minus the threshold value r_s . r_s specifies the allowable variation of a self point; a point at a distance greater than or equal to r_s from a self sample is considered to be abnormal. In this model, the radius is considered as a measure of the quality of the detector. Therefore, an effective detector must have an adequate radius, not cover self space, and not overlap with other detectors.

Negative Selection Algorithm for ship collision risk detection

3.1) Introduction

As we have discussed above, the importance of a collision risk detection method is immense for both the advance of manned and unmanned vessels. Marine researchers try to produce new reliable methods by utilizing the knowledge and the experience of seafarers and by developing new computational systems that can simulate the marine environment and the physics of a collision incident.

On an entirely different sector software engineers find ways to achieve goals by developing algorithms that imitate the human immune system. As shown, the application and the structure of these algorithms vary widely. For the purpose of this thesis, we focused on the negative selection algorithms and their applications on detecting anomalies within a system.

The idea behind this thesis was to combine the knowledge from two seemingly completely different sectors to produce a tool-method that can assist seafarers and marine researcher on detecting possible collisions. On the basis that the navigation of a ship is a process characterized by a number of factors-variables, we developed a negative selection algorithm that can detect anomalies in this process To be more precise, for our algorithm the safe navigation of own ship (OS) in relevance with a target ship (TS) is the “self space” of our feature space. The “non-self space” is the occurrences where the own ship (OS) finds itself in collision or a near collision course with the target ship (TS).

After defining the feature space, a negative selection algorithm was constructed based on the principles that were proposed by the researchers of Artificial Immune System. The main pillars of our work are the papers produced by Gonzalez [81] and Dasgupta [78], which contain fundamental steps of constructing a negative selection algorithm.

Next we used the work of collision risk detection researchers in order to define the variables that can adequately show the health state of the system. In this sector, the work of Perera and Soares [41] proved very valuable since they produced significant results by using easily obtainable factors from the own ship and the target ship.

The development of a negative selection algorithm for ship collision risk detection necessitates access to data comprising cases that culminate in collision as well as non-collision instances. However, acquiring such data is often challenging due to the infrequency and potential hazards associated with real-world collisions. Consequently, to surmount this limitation, a secondary simulation algorithm was devised. This auxiliary algorithm enables the generation of synthetic collision and non-collision scenarios, thereby providing a sufficient dataset for training and evaluating the negative selection algorithm.The subsequent phase of our work involved the development of a detector training method for our algorithm. This entailed designing a methodology for the algorithm to generate detectors that would be utilized for anomaly detection. Once a set of detectors was generated, we proceeded to develop a systematic approach for the algorithm to employ them towards accomplishing its objective. In addition, we conducted a sensitivity analysis to determine the optimal values for the primary parameters of the algorithm, which also provided valuable insights into the algorithm's behavior. Subsequently, we evaluated the accuracy and efficiency of our algorithm through testing and also conducted several case studies to demonstrate its practical applicability. The main steps of the process can be shown in the flowchart of Fig.12.

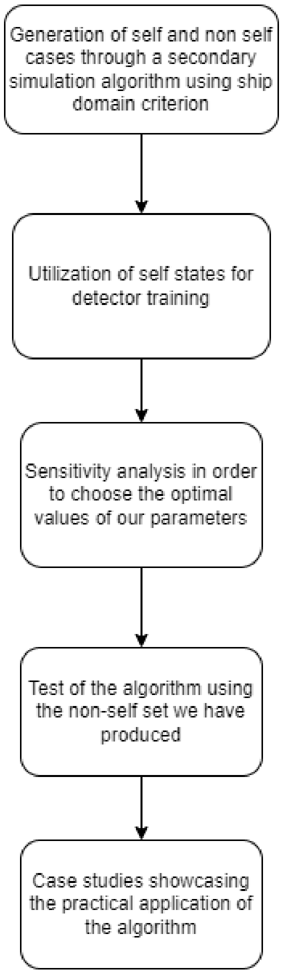

Figure 12: Main steps of the methodology that was applied

3.2) Dataset construction- Secondary algorithm

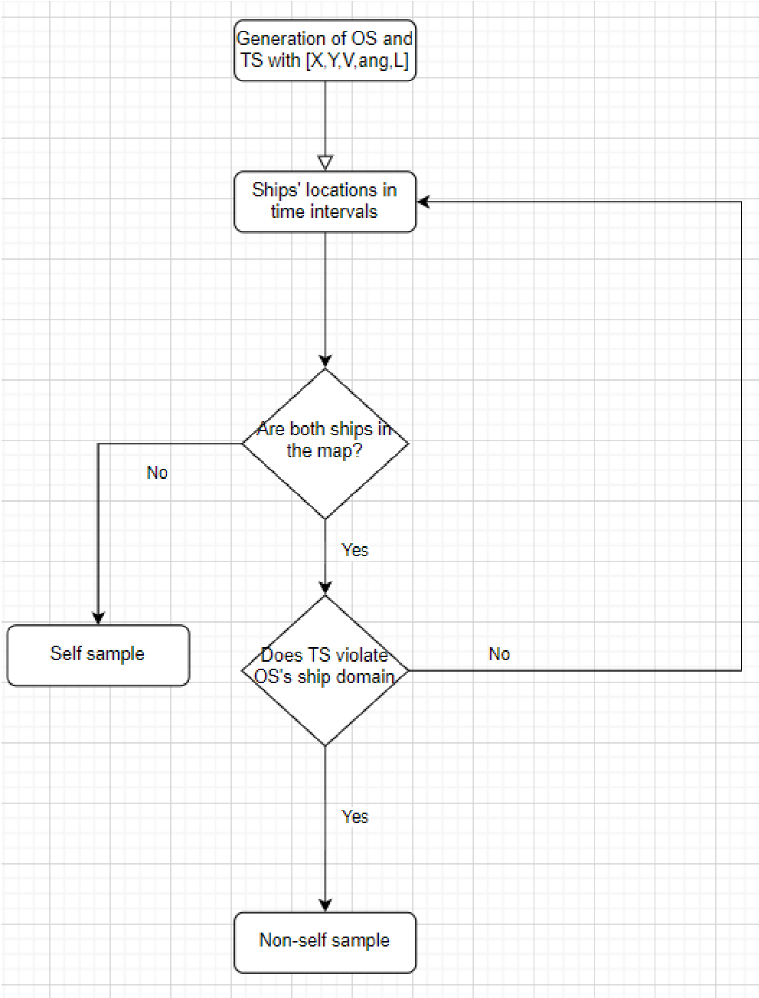

Figure 13: Flow chart of secondary algorithm

In order to produce the synthetic data we required for our algorithm we facilitated some of the techniques and methods that have already been used in collision avoidance studies. We developed an algorithm that simulates the movement of two ships (Own Ship and Target Ship) in a marine environment. As a next step we utilized an expert's based criterion, namely ship domain, in order to determine if a collision or a near-collision incident occurs for the vessels' simulated courses. Hence, the output of this algorithm are incidents where the two ships are in peril (non-self cases) and incidents where the two ships cross the marine area without causing harm to each other (self cases). As we mentioned in previous chapters, collision detection methods require the adoption of assumptions in order for the simulation of a marine environment to be possible. Below we will discuss the assumptions we made for the development of this algorithm. In Fig.13 we find a flowchart of the secondary algorithm.

3.2.1) Assumptions

3.2.1.1) Sea area

The marine space we chose has the following characteristics:

- It is two-dimensional. Therefore, the map is a layer based on Cartesian coordinates. We adopted this assumption since the physics of the collision are not affected by the z-axis morph of the vessel. This assumption is common in all the relevant research.
- There are no draft restrictions in the map. Therefore, the ships are free to navigate anywhere on the map without running aground.
- There are no physical objects in the subject area. This and the previous characteristic can define the subject sea area as an open deep sea.
- The marine space is a 4 km^2 area. More specifically we have designed a Cartesian coordinate system with X and Y axis, with $0 \leq x, y \ \leq 4000 \ (m)$

3.2.1.2) Ships' Dimensions and Velocities

The secondary algorithm generates scenarios for ship collision risk detection, where each scenario consists of two ships: an "own ship" and a "target ship." The ship types for both vessels are randomly selected from predefined ranges of ship types, and the characteristics of each ship, such as length (L) and velocity (V) are also randomly determined based on its respective type. This randomization process ensures diverse and realistic scenarios, encompassing various ship types and their associated characteristics for comprehensive analysis and evaluation.. The types and their respective ranges are the following:

- Bulk carrier. The length of a bulk carrier is drawn from the range [100m, 270m]. Its velocity is drawn from the range [10 knots, 18 knots].
- Oil tanker. The length of a tanker is drawn from the range [200m, 340m]. Its velocity is drawn from the range [8 knots, 16 knots].
- Containership. The length of a containership is drawn from the range [250m, 450m]. Its velocity is drawn from the range [15 knots, 27 knots].
- Fishing vessel. The length of a shipping vessel is drawn from the range [30m, 100m]. Its velocity is drawn from the range [5 knots, 17 knots].

All the values for the above ranges are based on internet data and on Papanikolaou [83].

Furthermore, it should be mentioned that the algorithm picks randomly the type of a ship and its velocity and length by assuming that the values in these ranges follow a uniform distribution. The utilization of a uniform distribution is motivated by its unbiased and fair nature, ensuring that each value within the range has an equal probability of being selected. This approach guarantees fairness in the random selection process and maintains integrity in generating diverse ship characteristics for the scenarios.

3.2.1.3) Ships' Trajectories

The algorithm simulates the trajectories of the OS and the TS by assuming the following:

- The courses of the ships involved in the scenarios are linear, reflecting a simplifying assumption where the paths of the ships are considered to be straight lines.
- The ships do not alter their velocity or course while in the area of research.

3.2.1.4) Other

Weather conditions and visibility are not taken into account and therefore they don't count as a parameter for the simulation.

The ships can be depicted as points on the map. This assumption can be taken since the safety criterion we chose, and which will be explained later develops a "safe area" for the ship which is much larger than the ship's area.

3.2.2) Simulation part of the algorithm

3.2.2.1) Case generation

As we mentioned above the secondary algorithm's first step is to generate a set of two vessels (OS and TS). We chose to study cases involving two vessels as most collision risk detection methods do. The study of more than two vessels moving through the map is only meaningful in cases where we study collision avoidance techniques. Since the purpose of this thesis is to produce a collision risk detection method, there is no reason for adding more than two ships.

The roles of OS and TS are assigned randomly to the two vessels generated. Each of the generated vessels has the following characteristics:

- Length. As we mentioned before each vessel is assigned to a ship category, which in turn provides the algorithm with the vessel's length. The length is utilized in the collision risk detection criterion used in the secondary algorithm. This criterion will be analyzed in 3.2.2.3
- Velocity. In the same way a certain velocity is assigned to each vessel. As we mentioned this velocity is indicative of the vessel's type and does not change while the vessel moves through the map.
- Course. The generated vessel gets an initial course angle that indicates its direction.
- Initial position in X axis (latitude) $Xo_{(t=0)}$ and initial position in Y axis (longitude) $Yo_{(t=0)}$ ..

The algorithm incorporates the random selection of initial positions and course angles for the ships, further enhancing the variability and authenticity of the generated scenarios. The values for these are chosen from predefined ranges. For latitude (Xo) and longitude (Yo) we assume that the range is [0,4000m] because of the size of the map we study. For the angle we assume that the range is [-π,π]. The random process follows the uniform distribution. The utilization of a uniform distribution is motivated by its unbiased and fair nature, ensuring that each value within the range has an equal probability of being selected. This ensures that a great variety of scenarios is generated.

To sum up after the case generation each vessel can be identified by the following matrices:

$$Own\ Ship = [Xo_{(t=0)}[m], Yo_{(t=0)}[m], Vo[kn], \varphi_o[rad], Lo[m]]$$

$$Target\ Ship = [Xa_{(t=0)}[m], Ya_{(t=0)}[m], Va[kn], \varphi_a[rad], La[m]]$$

The result is the initial positions as well as the speed and direction for two vessels. Below we will examine how the trajectory of the ships and the safety criterion are calculated.

3.2.2.2) Trajectory calculation

Since the main assumption of this algorithm is that the course and the velocity of the ships will remain unaltered while they move in the map, the algorithm can find the positions of the ships at any given time. As a first step, the velocity along with the course angle produce the axial velocities for each of the vessel as shown in the following equations.

$$V_{ox} = V_o[kn] * 0.5144 * \cos(\varphi_o) \ [{}^{m}/_{s}]$$

$$V_{oy} = V_o[kn] * 0.5144 * \sin(\varphi_o) \ [{}^{m}/_{s}]$$

$$V_{ax} = V_a[kn] * 0.5144 * \cos(\varphi_a) \ [{}^{m}/_{s}]$$

$$V_{ay} = V_a[kn] * 0.5144 * \sin(\varphi_a) \ [{}^{m}/_{s}]$$

As its next step the algorithm projects the future positions of the ships at regular time intervals (dt). The time interval in our simulation is dt=0.1s.

$$Xo_{(i)} = Xo_{(i-1)} + V_{ox} * dt$$

$$Yo_{(i)} = Yo_{(i-1)} + V_{oy} * dt$$

$$Xa_{(i)} = Xa_{(i-1)} + V_{ax} * dt$$

$$Ya_{(i)} = Ya_{(i-1)} + V_{ay} * dt$$

The simulations are executed until either the collision criterion is met or one of the ships navigates outside the spatial boundaries of the map being analyzed. Depending on which exit condition is triggered first, the initial scenario is classified into one of two groups, namely self-sample or non-self-sample. A scenario is considered to be in the self-sample if both ships successfully navigate the map without encountering any imminent collision threat. Conversely, if the collision criterion is satisfied at any point during the simulation, the scenario is classified to be in the non-self-sample. In the following section, we will elaborate on the specific collision criterion that we have selected for our algorithm.

3.2.2.3) Collision criterion

In on our study, we chose to integrate a commonly used expert-based method in order to build the dataset for our main algorithm. This method is the ship domain method and more specifically the version proposed by Coldwell [45], where the ship's domain is represented by an elliptical region. The center of the ellipse is at the center of the ship, with a semi-major axis of 6 times the length of the ship (L) and a semi-minor axis of 1.75 times L (Fig.14) . The violation of the own ship's (OS) domain by the target ship (TS) is regarded as the collision criterion for the algorithm. During the phase of case generation for the secondary algorithm, all the essential values for generating the ship domain were produced. To determine whether a collision risk is present, the algorithm checks in each time interval whether the target ship falls within the own ship's domain.

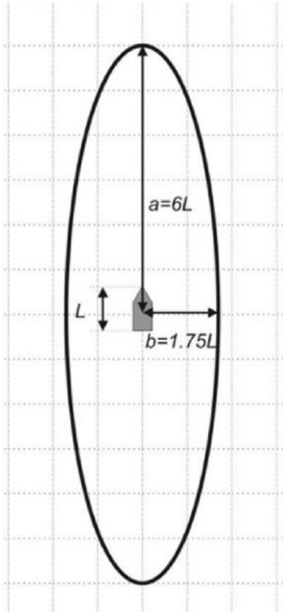

Figure 14: Coldwell's ship domain, [45]

3.2.3) End of the secondary algorithm

The algorithm operates in a loop continuously as described until a predefined exit condition is reached. In this case, since the purpose of the algorithm is to build a database that will be utilized in the primary algorithm, the exit condition is based on the number of self and non-self samples required. The self samples are used during the detector training phase, which will be discussed in later sections, while the non-self samples are utilized to evaluate the efficacy of the mature detectors produced.

3.3) Negative selection algorithm

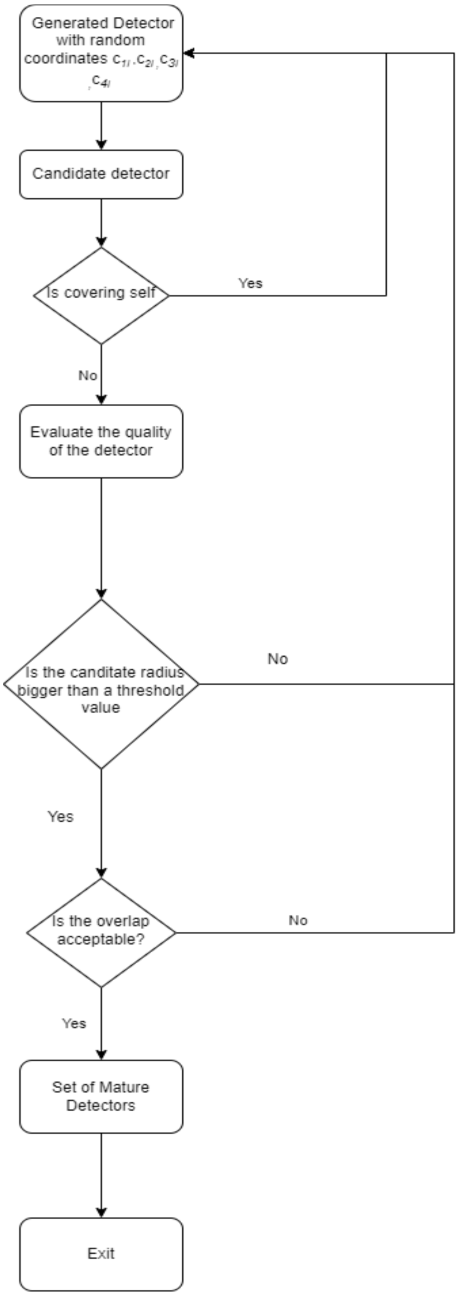

Figure 15: Detector training phase flow chart

3.3.1) Definition of the main parameters

The number of health-indicating parameters is a critical factor in the development of an effective negative selection algorithm. If the number of parameters is too small, the resulting model may not provide a complete understanding of the system being studied. Conversely, if the number of parameters is too large, the detector training phase can become overly complex. Additionally, since each parameter is considered an equal contributor to the overall health of the system, the inclusion of a large number of irrelevant factors can diminish the significance of the main factors.

In our cases after taking the former into consideration, we chose to incorporate the knowledge of other researchers into our method. In 2015, Lokukaluge Perera and Guedes Soares produced a study called “Collision risk detection and quantification in ship navigation with integrated bridge systems” [40]. The aim of that study was to develop a collision detection methodology that takes into account uncertainties in vessel state during complex maneuvers, as part of an integrated bridge system. The paper discusses various modern technologies that can be implemented in integrated bridge systems to enhance navigation safety during close encounters between vessels. Furthermore, ship navigation tools that can identify potential collision situations in advance are introduced. Their methods are evaluated under a two vessel encounter in a collision or near-collision situation. The primary benefit of this method is its ability to generate valuable and informative results using basic and readily available information from the involved ships. The primary parameters used were the relative axial distances and velocities between the ships.

The notion that a risk detection method can be produced by easily obtainable factors thus eliminating uncertainties regarding the target ship’s movement, drove us to try and produce our method by utilizing the same parameters as Perera and Soares. A health indicating vector has the following structure:

$$[X_{OA}, Y_{OA}, Vx_{OA}, Vy_{OA}]$$

X_{OA}=$X_A - X_O$ with X_A , X_O the positions of the target and own ships’ centers in the x-axis respectively

Y_{OA}=$Y_A - Y_O$ with Y_A , Y_O the positions of the target and own ships’ centers in the y-axis respectively

Vx_{OA}=$Vx_A - Vx_O$ with Vx_A , Vx_O the axial speed of the target and own ships in the x-axis respectively

Vy_{OA}=$Vy_A - Vy_O$ with Vy_A , Vy_O the axial speed of the target and own ships in the y-axis respectively

In the following chapter we will discuss how we preprocessed our dataset as the first step of our algorithm.

3.3.2) Dataset pre-processing

As previously described, the dataset utilized in this algorithm comprises synthetic data generated using a secondary algorithm (Described in Chapter 3.2). The dataset used in this study contains positional data for two ships, along with their respective velocities and courses at a specific time. The dataset is divided into two sets: cases involving a collision risk and cases that conclude with the safe passage of the ships. This data will be processed to generate health-indicating vectors for the system under study. Each case produces a vector that can be classified to be either in the self sample or in the non-self sample and it takes the format of the vector we mentioned in the previous chapter.

In order to prepare the data for analysis, it is important to normalize the health indicating parameters using a process commonly used in negative selection algorithm research.

Normalization ensures that each parameter falls within the interval [0,1], creating a set of data points within $[0,1]^N$, where N is the number of parameters in the system. This allows each parameter to contribute equally to the analysis, regardless of differences in value range.

The concept of normalization is often used in machine learning and data analysis to scale and transform data to a common range, which can help to prevent certain features from having undue influence on the results of the analysis. In the context of negative selection algorithms, normalization can be particularly important as it ensures that all parameters used to describe the system have an equal opportunity to contribute to the health indicator vector. Additionally, normalization can help to improve the performance of the algorithm by reducing the impact of outliers or data that may be disproportionately large or small.

For the normalization of our data, we must first identify the range of each parameter, i.e. the minimum and maximum values. Then we subtract the minimum value from each parameter value to shift the range so that it starts at zero. As a final step we divide each parameter value by the range (i.e., the difference between the maximum and minimum values) to scale it to the range [0,1].

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

By normalizing the health indicating parameters of the system to the interval [0,1], negative selection algorithms have the ability to translate complex ship collision risk detection problems into the study of a multi-dimensional unit cube, $[0,1]^N$, where each dimension represents a different health indicating parameter. This transformation simplifies the problem and allows for easy comparison and classification of data.

After the end of the preprocessing our new dataset is separated into the self set and the non-self set. Each element of these sets is an array with the following form, and it can be visualized as a point in a four dimensional cube.

$$[norm\_X_{OA}, norm\_Y_{OA}, norm\_Vx_{OA}, norm\_Vy_{OA}] \in [0,1]^4$$

3.3.3) Detector training

The detector training phase involves the generation of a set of detectors that are designed to respond to non-self (anomalous) patterns in the system under study. In our case that translates to detectors that can find through simple information when two ships head towards a collision or a near collision event. We based our detector training method on the approach of Dasgupta and Gonzalez [78, 81]but with some deviations from their method.

3.3.3.1) Self radius

The aforementioned approach suggests that each self point defines a self-hypersphere, which is a region that has the self point as its center and a radius that defines its size called self radius (r_s). Each point that is found within the hypersphere can also be defined as a self point. The mathematical definition of the set of points in the self hypersphere is shown in the following equation.

$$\sqrt{\sum_{i=1}^{i=4}(x_i - s_i)^2} \leq r_s$$

The square above is the Euclidean distance between a point X within the hypersphere and the self point S at its center.

In the development of our negative selection algorithm, we chose a fixed value to represent the self radius. The self radius is a crucial parameter in the algorithm and its value significantly affects the performance of the algorithm. A small self radius may result in reduced sensitivity to the detection of other self points, thereby increasing the likelihood of false positives, whereas a large self radius may lead to false negatives, i.e., anomalies being incorrectly classified as normal data points, and hence, lower the accuracy of the algorithm. To determine the optimal self radius value for our algorithm, we conducted a sensitivity analysis on the main parameters and selected the value that yielded the best performance results.

3.3.3.2) Detector form

Since a set of self points has been produced thanks to the secondary algorithm, their respective self hyperspheres can also be defined in the $[0, 1]^4$. Any point that is not covered by the self-hyperspheres is considered a potential non-self point. In order to cover these areas, new hyperspheres are generated to serve as detectors for the system. These detectors can be expressed as follows:

$$D_i = [d_{i1}, d_{i2}, d_{i3}, d_{i4}, r_{di}]$$

Each detector has as its center a point of $[0, 1]^4$ hypercube and is linked with a value r_{di} which is called detector radius and is the radius of the hypersphere created by each detector. These hyperspheres define the capability of detectors to identify non self points. The mathematical expression of the set of points detected by a detector is shown in the following equation.

$$\sqrt{\sum_{j=1}^{j=4}(x_j - d_{ij})^2} \leq r_{di}$$

3.3.3.3) First filter-Detector radius

In the algorithm that we have developed, each detector has a distinct radius. As the primary function of the detector is to detect non-self instances, it is imperative that the hyperspheres of the detectors do not intersect with the hyperspheres of the self-instances. Thus, a precondition for determining the radius of each detector is established, whereby the distance between the center of the detector and its closest self-point must be equivalent to the sum of the radii of the detector and the self-point. As the detector radius is variable in our case, we have employed this precondition to calculate the radius of each detector as shown in the following equation.

$$r_d = Dis(D, S) - r_s = \sqrt{\sum_{i=1}^{i=4}(d_i - s_i)^2} - r_s$$

With S being the self point closest to the detector we study.

During the detector training phase, candidate detectors are randomly generated. The radius of each candidate detector is calculated by calculating its distance from all self points using the Euclidean distance metric. If any of these distances is smaller than the self radius r_s, the candidate detector is deemed unsuitable and a new detector is generated. Conversely, if all calculated distances are greater than the self radius, we proceed to compute the candidate detector's radius. This radius is determined by subtracting the self radius from the smallest calculated distance, which represents the distance between the candidate detector and its closest self point.

After the candidate detector's radius has been calculated, the first suitability criterion-filter is applied. According to that, the detector must be of an adequate size in order to belong to the set of mature detectors. This means that the calculated radius must be greater than a threshold value we choose. If the detector radius is smaller than the threshold value, the candidate detector gets rejected and we proceed with the next generated detector.

The detector radius is a critical parameter in the negative selection algorithm, as it determines the region around each detector that is deemed acceptable for anomaly detection. A detector with a small radius may result in a high false negative rate, as anomalies that fall outside the radius may go undetected. Conversely, a detector with a large radius may lead to a high false positive rate, as normal data points that fall within the radius may be falsely classified as anomalies. Therefore, it is crucial to set the detector radius appropriately to ensure the optimal balance between detection sensitivity and specificity. To achieve that we performed a sensitivity analysis to determine the optimal threshold value.

3.3.3.4) Second filter – Detector overlap

After passing the initial filter, a candidate detector is obtained, which encompasses a non-self hypersphere with a detector radius greater than the threshold value. The subsequent filter examines the overlap between the candidate detector and the existing mature detectors.

Avoiding a big overlap between detectors is crucial in the negative selection algorithm, as it helps to prevent redundancy and improve detection accuracy. When detectors have a large overlap, they may redundantly cover the same region of the data space, leading to increased computational cost and decreased detection performance. Furthermore, a key objective of a successful negative selection algorithm is to ensure that a non-self data point is detected by only one detector. This exclusivity allows for the subsequent characterization of a detector based on the non-self cases it accurately identifies. Therefore, it is important to carefully design the overlap constraint in the negative selection algorithm to ensure that the detectors are sufficiently distributed and cover the data space effectively, while avoiding excessive overlap that could compromise the algorithm's performance. In Fig.16 we can find a depiction of what overlap means [82].

To achieve a small overlap for our detectors, we adopted the overlapping criterion developed by Dasgupta et al in their research around negative selection algorithms in aircraft fault detection [78]. According to this criterion, an overlapping measure W for a detector is calculated as follows:

$$W(d) = \sum_{d \neq d'} w(d, d')$$

Where $w(d, d')$ is the measured overlap between two detectors d=[c, r_d] and $d' = [c', r_{d'}]$

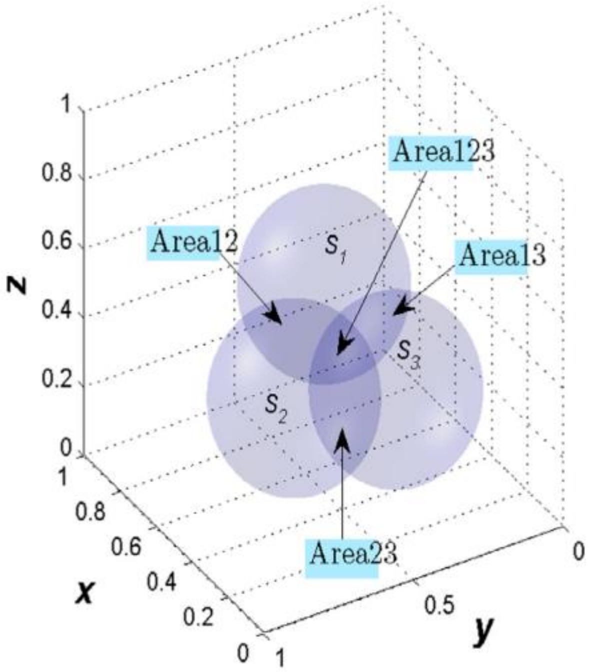

Figure 16: Overlap between detectors, [82]

$w(d, d')$ is measured as $w(d, d') = (\exp(\delta) - 1)^m$, where m is the dimension of the feature space (in our case m=4) and δ is calculated according to the following equation.

$$\delta = (\frac{r_d + r_{d'} - D}{2r_d})$$

The value of δ is bound between 0 and 1 and D is the distance between the centers of the detectors. According to the criterion, $W(d)$ must not exceed a threshold value ξ. The measure of overlap appears to give an advantage to detectors with larger radii, that is, detectors that cover a larger portion of the non-self space with minimal overlap between them. As a result, the closer the centers of two detectors, the greater the value of the overlap measure $w(d, d')$. The threshold value was obtained through trials. In Fig.17 we can see a depiction of the main parameters of Dasgupta's method.

During the initial iterations of the algorithm, the criterion is not applicable, as it lacks meaning in the absence of mature detectors. After we have obtained four mature detectors , the filter is implemented, and if a candidate detector passes the overlap test, it is considered a mature detector and added to the corresponding set. Otherwise, the detector is rejected, and the next candidate is evaluated.

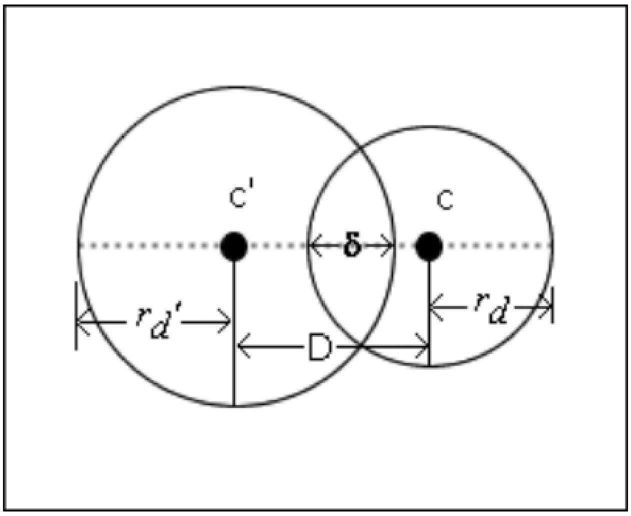

Figure 17: Schematic illustration of Dasgupta's method, [78]

3.3.3.5) Exit condition of detector training – Number of detectors

The detector training process is iterative and finishes when a predefined number of mature detectors is obtained.

The total number of mature detectors in a negative selection algorithm is an important parameter that can have a significant impact on the algorithm's performance. The purpose of negative selection algorithms is to detect anomalous patterns in data, and the number of detectors determines the level of sensitivity and specificity of the algorithm. Having too few detectors may result in high false positive rates, while having too many detectors may result in a high false negative rate [82]. Therefore, selecting an appropriate number of detectors is crucial to ensure that the algorithm is both effective in detecting anomalies and efficient in processing large amounts of data. A well-tuned number of detectors can enhance the algorithm's ability to distinguish between normal and abnormal patterns, thereby improving its accuracy and reliability in various applications such as intrusion detection, anomaly detection, and fault diagnosis.

In addition to the impact on performance, it is also important to note that a large number of detectors can increase the time required to train the detector in a negative selection algorithm. The process of training detectors involves generating a large number of random patterns and computing their distances from the set of self-patterns, which can be computationally intensive. As the number of detectors increases, so does the number of random patterns and the computational complexity of the training process. Therefore, it is important to balance the number of detectors with the available computational resources to ensure that the training process is efficient and does not become a bottleneck in the overall algorithm performance.

We conducted a sensitivity analysis to determine the optimal number of detectors, along with other detector training parameters. The details of the sensitivity analysis will be presented in a subsequent section.

3.3.4) Algorithm process-“Antigen” detection

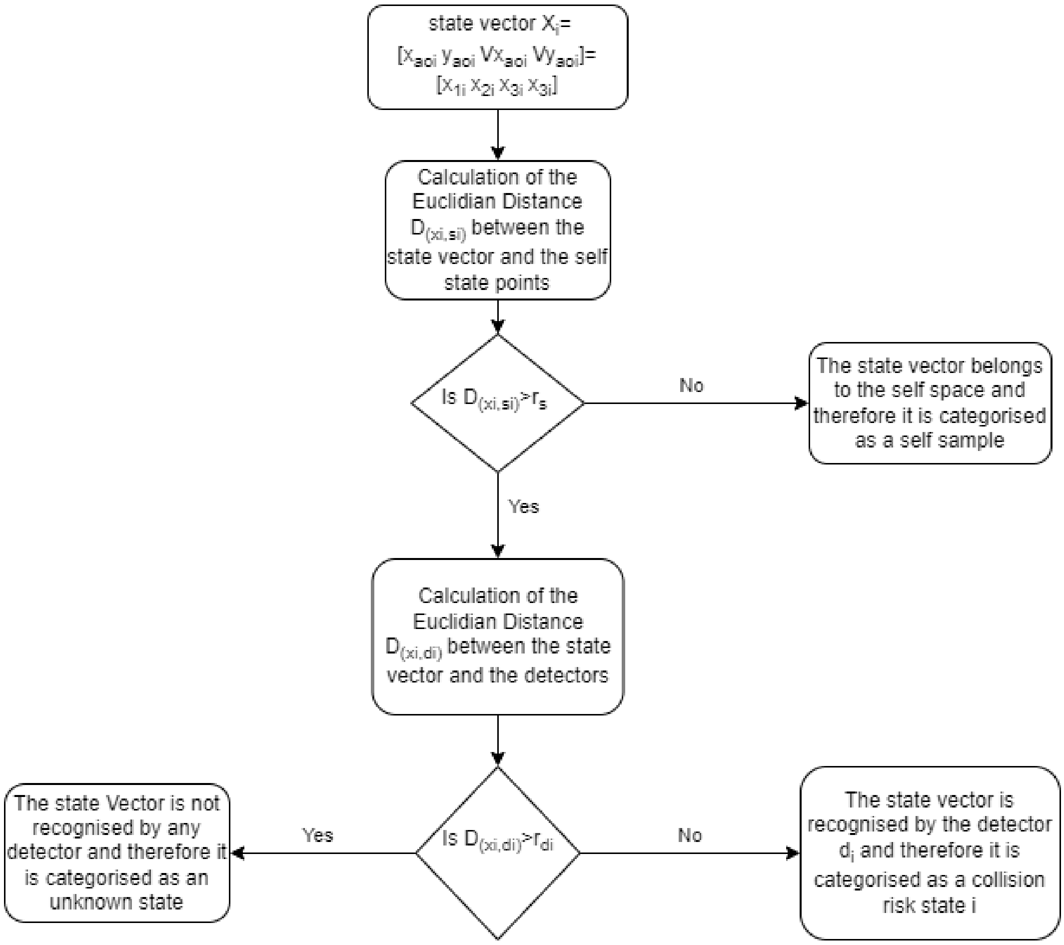

Figure 18: Negative Selection Algorithm’s flow chart

The next step after producing a set of mature detectors is to evaluate their quality. Before doing so, we will outline the key features of our detection algorithm and describe its functionality. In figure 18 a flow chart of the algorithm in its present version is presented.

As previously discussed, a negative selection algorithm is designed to identify anomalies within a system. In our case, the system under consideration is the navigation of two ships in open sea. A safe and healthy state of the system occurs when both ships navigate through the area of interest without any collision risk, which can be defined as a "self" state. Conversely, any situation that poses a risk of collision between the two ships is considered a "non-self" state of the system. By utilizing a negative selection algorithm, we can detect any potential collision risks, thereby helping to ensure the safe passage of both vessels.

As previously discussed, we have determined that the inputs required to study the system are easily obtainable and consist of the following: the location of the own and target ships in a two-dimensional system, their respective velocities, and their course angles. These parameters are processed in order to produce the health indicators of the negative selection algorithm. These indicators are the relative axial distances and velocities for the vessels. As we mentioned though the inputs of the algorithm must be normalized.

It becomes obvious that every case that might occur in the map we study can be transformed to an input vector of the following form.

$$[norm\_X_{OA}, norm\_Y_{OA}, norm\_Vx_{OA}, norm\_Vy_{OA}] \in [0,1]^4$$

Once this form is obtained, the algorithm comes in place in order to detect if the case contains a collision risk. The particulars of each step of the algorithm will be discussed below.

3.3.4.1) Check if the vector is recognized as a self-state

As previously mentioned, the negative selection algorithm operates in a four-dimensional hypercube defined by the range [0,1]. Within this hypercube, the self states that have already been defined generate self-hyperspheres - specific areas within the hypercube consisting of points that can be defined as self-states.

It becomes obvious that in the same way a detector can locate non-self points (cases), the identified self cases of the system can locate other self samples. Initially, the algorithm checks the case in question against our self dataset. The Euclidean distances between the case-point in question and the self samples are measured. Each calculated distance is then compared to the self radius r_s (radius of the self hypersphere). If the distance is smaller or equal to the self radius, the case-point in question is characterized as a self case. If it is bigger it can't be detected by this self point and the next distance is calculated.

Once it is ensured that no self point can detect the case in question, the algorithm proceeds with the utilization of the mature detectors dataset.

3.3.4.2) Check if the vector is recognized by a detector

To determine whether the case under study is a non-self state, the algorithm assesses whether the case vector is located within a detector's hypersphere. To achieve this, the algorithm calculates the Euclidean distance between the point in question and the center of a given detector. It then compares this distance to the detector's radius (r_{di}). If the distance is equal to or less than the radius, the algorithm identifies a potential collision risk. The detector responsible for detecting this risk then characterizes the type of collision risk involved. If the distance exceeds the radius, the algorithm proceeds to the next detector.

If the case vector is not identified by any detector, it is characterized as an unknown state. Although addressing unknown states in the algorithm is out of the scope of this thesis, potential ways of enhancing the adaptability of the model will be discussed in a subsequent chapter.

Sensitivity Analysis – Efficiency of the algorithm

The choice of parameters in negative selection algorithms (NSAs) has been a subject of discussion and debate in the research community [82]. The self radius, detector radius, and number of detectors are important parameters in NSAs that significantly affect the algorithm's performance. The self radius establishes the size of the self space, which is the region surrounding each self that is excluded from the detector space. The detector radius determines the range of detectors capable of detecting a given antigen. The number of detectors is a key determinant of the diversity of the generated detector population. Yet the method of choosing these factors is in most cases not explicitly described.

Chen Wen and Li Tao [82] conducted a parameter analysis of NSAs to investigate the impact of different parameter values on the algorithm's performance. They utilized a benchmark dataset to evaluate the performance of the algorithm under different parameter settings. Their analysis provides valuable insights into the optimal parameter settings for NSAs. Their research highlights the significance of meticulously selecting parameters in NSAs and provides guidance on how to achieve this effectively. By comprehending the consequences of different parameter settings, researchers can develop more effective and efficient NSAs that are better suited for their specific application domain.

Moreover, the metrics used to evaluate the effectiveness of negative selection algorithms (NSAs) can differ considerably. Typically, studies employ the accuracy percentage, which represents the proportion of identified cases from the input dataset that are correctly classified by the algorithm. This method was also used in this thesis.

In our approach, we conducted a series of trials to determine the optimal parameter values that produced the best results. The effectiveness of the algorithm was measured by two metrics: the accuracy percentage and the total time required for the algorithm to complete both the detector training and the identification of cases within the input dataset.

The parameter selection process involved the following steps. Initially, we performed a number of trial runs to establish a base combination of the main parameters, including the self radius, the minimum detector radius, and the number of detectors generated. This base combination was used as a starting point for subsequent trials, in which we systematically varied one parameter at a time while keeping the other parameters constant.

This method enables a systematic approach towards optimizing the algorithm's performance, by striking a balance between accuracy and practical constraints such as computational resources and time. Additionally, this method provides valuable insights into the influence of each parameter on the algorithm's outcome, thereby facilitating a sensitivity analysis. Therefore, this method is an effective way to improve the performance of the algorithm while simultaneously gaining a better understanding of its behavior.

The iteration method is executed in the following manner: one parameter is selected for analysis, while all other parameters remain constant at their base values. The selected parameter's value is systematically adjusted in incremental steps by a constant factor. Since the detector generation process incorporates randomness, the algorithm is executed ten times for each value of the selected parameter. For each iteration, the accuracy percentage and the total time required for detector training and case evaluation are measured. The accuracy percentage and total time values are computed as the median of the ten iterations for each parameter value.

To calculate the accuracy percentage, 30 non-self cases generated by the secondary algorithm are provided as input to the algorithm for each iteration. Since each iteration involves a distinct set of mature detectors, the algorithm's performance is evaluated using the number of non-self cases detected correctly relative to the total number of inputs, according to the following equation.

$$accuracy = \frac{detected\ cases}{input\ cases}$$

The total time required for each iteration is determined by a module within the program's language environment. This time is comprised of two components: the duration of the detector training process, and the time taken by the algorithm to evaluate the input cases.

In the following chapters we will present and discuss the results of the sensitivity analysis.

4.1) Self radius

As we mentioned the self radius is a vital parameter in the algorithm, and its value has a substantial impact on the algorithm's performance. A small self radius can reduce the sensitivity of the algorithm towards detecting other self points, leading to an increased rate of false positives. Conversely, a large self radius can result in false negatives, i.e., anomalies being misclassified as normal data points, thereby decreasing the accuracy of the algorithm. Moreover, the self radius parameter plays a critical role in determining the size of the self-hyperspheres and, consequently, has a direct influence on the algorithm's ability to generate mature detectors. This effect is observed in cases where the detector radius is either a fixed value or a minimum threshold, as in the current study.

In Fig.19 we can see a diagram depicting the relationship of accuracy percentage and self radius in our algorithm. In Fig.20 we can see the relationship between self radius and total execution time of the algorithm.

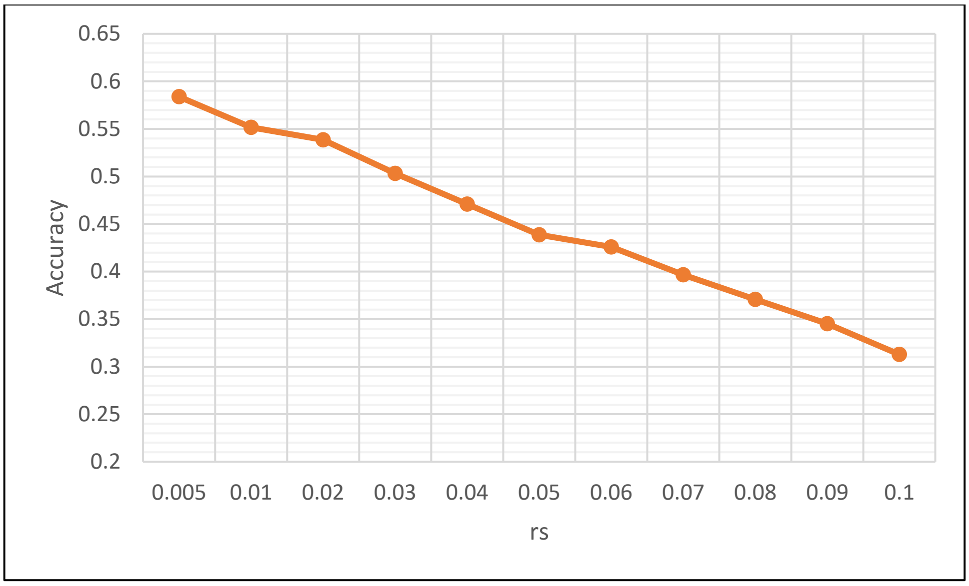

Figure 19: Accuracy percentage over different values of self radius

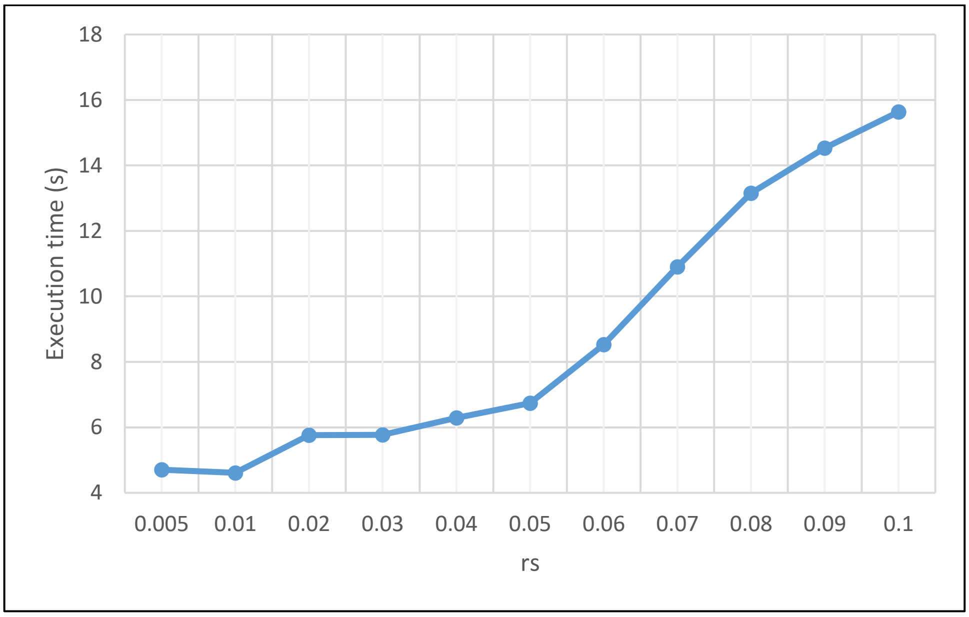

Figure 20: Total execution time of the algorithm over different values of self radius

The sensitivity analysis provides empirical evidence for the theoretical deductions made above. It shows that a smaller self radius results in more efficient and faster detector training. Increasing the self radius leads to a steady decrease in accuracy and a significant increase in the total execution time. Based on these results, we conclude that a self radius of 0.01, the base value used in this study, is appropriate for the algorithm.

4.2) Minimum detector radius

As we mentioned in the detector training, minimum detector radius is used as a criterion in the development of mature detectors. If the generated detector has a detector radius smaller than the threshold radius it is discarded. This criterion was included in our algorithm due to

Dasgupta's [78] notion that detectors with a bigger radius should be favored in the mature detector dataset. However, it is important to note that an excessively large detector radius can lead to false positive results and increase the complexity of the training procedure. This is due to the fact that our algorithm has a predefined number of mature detectors along with an overlap criterion. Therefore, setting an appropriate detector radius is crucial for balancing the trade-off between sensitivity and specificity in anomaly detection.

In Fig.21 we can see a diagram depicting the relationship of accuracy percentage and minimum detector radius in our algorithm. In Fig.22 we can see the relationship between minimum detector radius and total execution time of the algorithm.

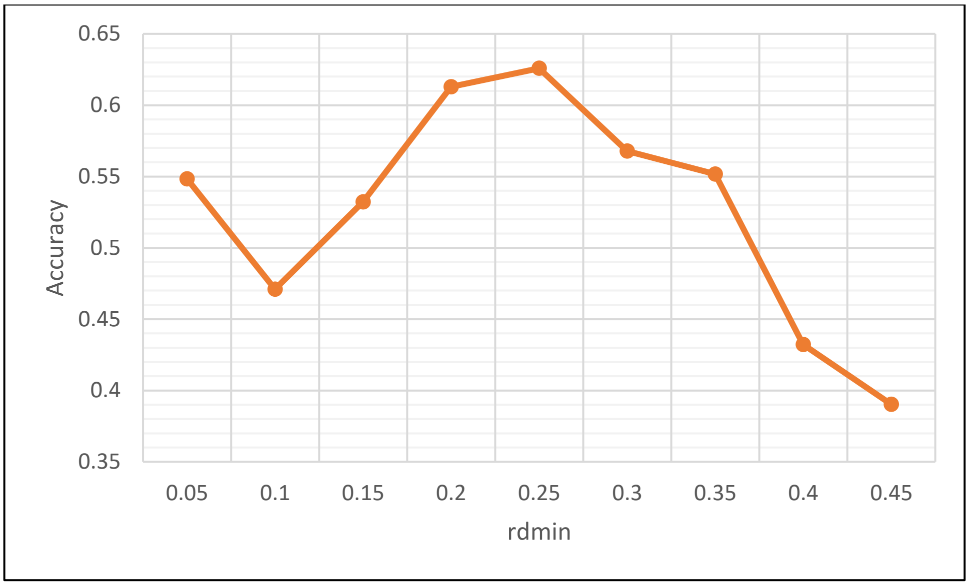

Figure 21: Accuracy percentage over different values of threshold detector radius

The sensitivity analysis results confirm the theoretical deductions from our work and Dasgupta's research. Initially, increasing the threshold value improves detector accuracy with negligible impact on total execution time. However, after a certain point, further increases decrease algorithm accuracy and significantly increase execution time. This is because it becomes increasingly challenging to cover non-self areas with excessively large detectors, and generating mature detectors that do not overlap with existing ones becomes more difficult.

Based on the sensitivity analysis, we have determined that a benchmark value of 0.3 produces fair results in terms of accuracy and execution time. However, we also suggest that the benchmark value could potentially be moved towards 0.25 for even better performance.

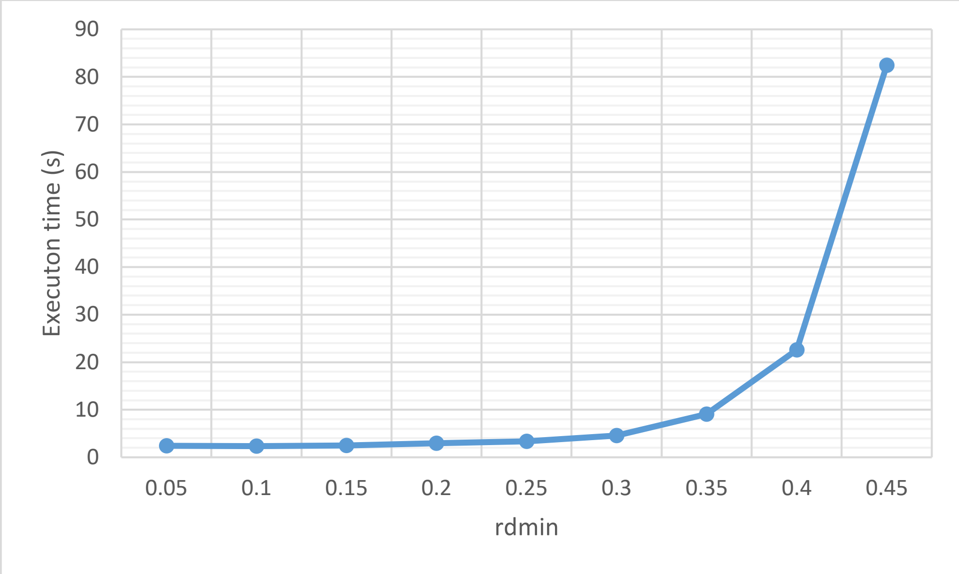

Figure 22: Total execution time of the algorithm over different values of threshold detector radius

4.3) Number of detectors

The number of detectors is a crucial factor in the performance of the algorithm as it directly affects the algorithm's detection sensitivity and specificity. A higher number of detectors can lead to better coverage of the search space, increasing the sensitivity of the algorithm to detect anomalies. However, a larger number of detectors may also result in more false positives, lowering the specificity of the algorithm. Furthermore, a large number can greatly increase the execution time.

In Fig.23 we can see a diagram depicting the relationship of accuracy percentage and the size of the mature detector dataset. In Fig.24 we can see the relationship between the size of the mature detector dataset and total execution time of the algorithm.

The impact of the number of detectors on the negative selection algorithm's performance is evident in the results, as an increase in the number of detectors leads to an increase in accuracy. However, it should be noted that the rate of accuracy improvement decreases as the number of detectors increases, and the execution time significantly increases with larger datasets. Therefore, selecting the optimal number of detectors requires considering the computational capabilities of the system running the algorithm.

For our specific implementation, we initially chose a base value of eighty detectors, but based on the results of our sensitivity analysis, we concluded that increasing the number of detectors to ninety would result in a more accurate algorithm without significantly increasing the total execution time.

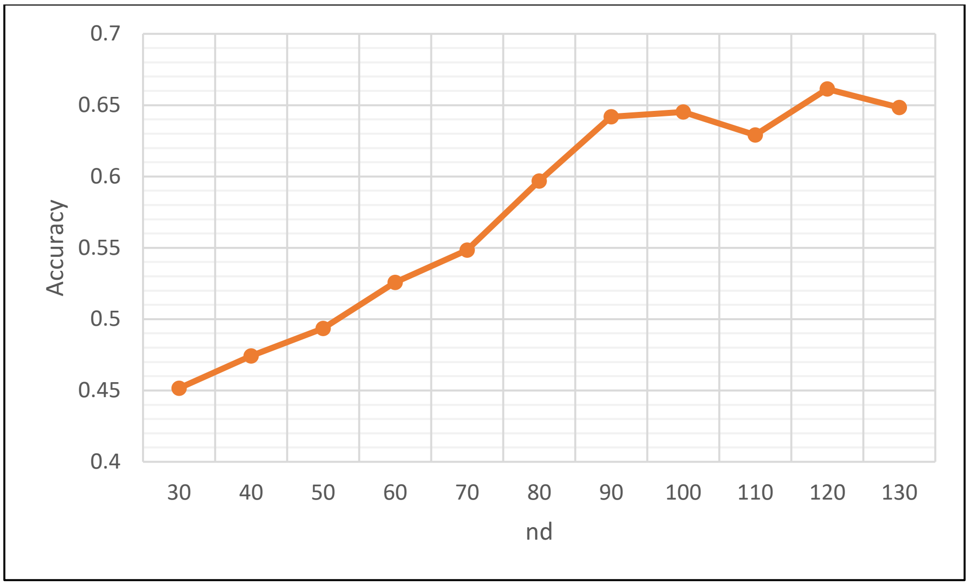

Figure 23: Accuracy percentage over different number of detectors

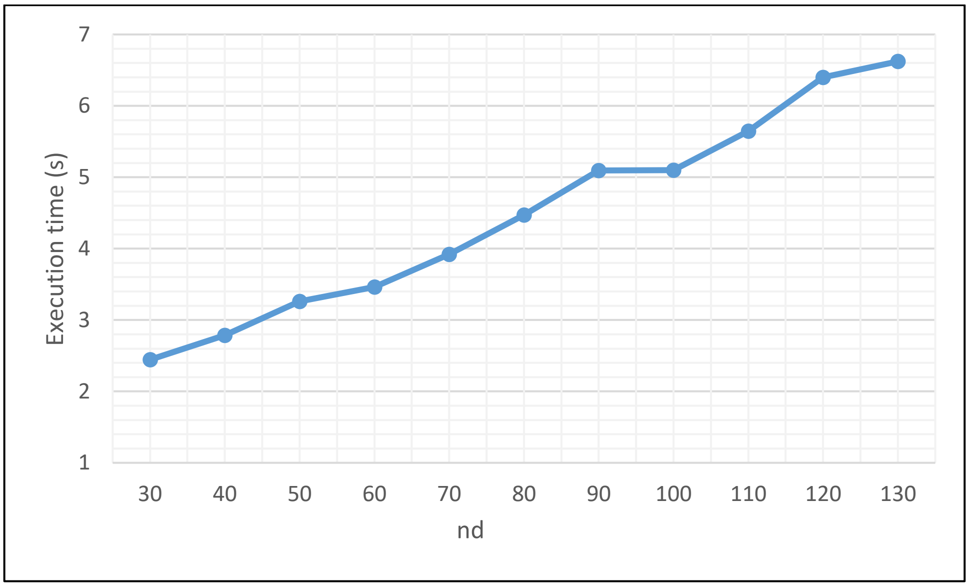

Figure 24: Total execution time of the algorithm over different number of detectors

Results

The sensitivity analysis served a dual purpose in our research. Firstly, it provided valuable insights into the effects of changes in the main parameters on the efficiency and accuracy of the algorithm. Secondly, it enabled us to identify the optimal values for these parameters and define our algorithm accordingly.

The values that we chose are the following. The self radius was picked as 0.01. The detector radius threshold value 0.25 with ninety (90) mature detectors. Table 1 shows the results of ten iterations of the algorithm. In each iteration ninety detectors are trained using 60 self samples and are used to detect a set of 30 non-self cases.

The average accuracy of the algorithm is 65,81% and the whole process lasts on average 3.4267 seconds. The non-self cases not detected by the algorithm are defined as unknown cases.

Table 1, The accuracy percentage and the total execution time for the selected parameters

rs=0.01,rdmin=0.25,nd=90				
a/a	t/n(accuracy)	time training(s)	time algorithm (s)	total time (s)
1	0.709677419	2.402990103	0.874857903	3.277848005
2	0.64516129	2.516611814	0.865339518	3.381951332
3	0.612903226	2.327028751	0.845599413	3.172628164
4	0.677419355	2.613990307	0.812807083	3.42679739
5	0.741935484	2.280589342	0.847968817	3.128558159
6	0.612903226	2.489983559	0.987318993	3.477302551
7	0.64516129	2.813468218	1.043375015	3.856843233
8	0.64516129	2.39804554	0.952609062	3.350654602
9	0.612903226	2.579602003	0.928194761	3.507796764
10	0.677419355	2.791289806	0.895800591	3.687090397
average	0.658064516	2.521359944	0.905387115	3.42674706

An essential feature of the developed algorithm is its remarkable accuracy in detecting potential collision situations without falsely characterizing non-self samples as self. This outcome is particularly crucial in scenarios where the algorithm is in use on board a ship. In these cases, any mischaracterization of a collision situation as safe could lead to catastrophic consequences. The results of our study demonstrate that the developed algorithm can prevent such errors from occurring, thereby enhancing the overall safety of maritime operations.

A second important comment about the results is that the algorithm achieves a very remarkable accuracy with limited computational resources. As a point of reference, Wen and

Tao [82] had computational capacity to produce numbers of detectors in the area of 10^{24} when our computational capacity can produce detectors in the area of 10^{2}. That tends to show great potential for the algorithm as a tool in collision risk detection.

Another noteworthy aspect of our algorithm is its efficient execution time. In roughly 3.5 seconds, the algorithm can train a number of detectors and identify a set of 30 individual cases. In a potential real-world application, the detector set would be trained only once, which means that the identification of an individual case would take only a fraction of a second, even with limited computational resources. This highlights one of the main advantages of our method, its speed, which could be a critical factor in real-world collision risk detection scenarios where swift action is required to prevent accidents.

In summary, our algorithm demonstrates a noteworthy level of accuracy and efficiency, with no instances of false identification of non-self cases. These findings suggest that our algorithm holds promise as a reliable method for detecting collision risks in ship navigation.

Moving forward, our conclusion chapter will provide a detailed analysis of the results, exploring their implications and addressing potential limitations. Additionally, we will consider opportunities for further development and enhancement of our algorithm.

In subsequent chapters, we will showcase several case studies to provide a more practical illustration of how our algorithm would operate in real-world scenarios. By doing so, we aim to demonstrate the potential value of our approach to improve maritime safety.

Overall, our study provides evidence that our algorithm offers a valuable contribution to the field of collision risk detection in ship navigation. By presenting a comprehensive analysis of our results and exploring opportunities for future research, we hope to contribute to ongoing efforts to enhance safety in maritime transportation.

In this chapter, we will explore several case studies to demonstrate the effectiveness of our negative selection algorithm in collision risk detection. As discussed earlier, the algorithm uses a self/non-self recognition mechanism to identify anomalous behavior that may lead to collisions. In the following sections, we will present two case studies that resulted in collisions and two that did not, where the algorithm's predictions were accurate. By analyzing these cases, we aim to provide a better understanding of the algorithm's capabilities and its potential for real-world applications.

The case studies were conducted using a customized script that emulated the behavior of our secondary algorithm. Specifically, the script generated pairs of ships, consisting of an own ship and a target ship, and placed them in the relevant map with their respective initial positions, velocities, and headings. Our algorithm was then applied to the data of each ship, which were processed using a detector or a self-hypersphere. Once the algorithm determined the state of each case, the simulated routes of the ships were generated by assuming that they would maintain their current velocity and heading. Subsequently, the simulation output was used to identify the closest point of approach between the two ships and determine if a collision was imminent or not.

6.1) Case 1 – Collision

In this case two containerships with the following characteristics were generated by the script.

Table 2, Case 1: Data of the pair of ships in the initial state

	Type	Length (m)	Initial X pos	Initial Y pos	Velocity (Knots)	Heading (rads)
Own Ship	Container	310	2805	2400	15	-0.79
Target Ship	Container	290	1847	1035	18	0.04

In this case study, two container ships were modeled moving eastward on a given map. The algorithm was applied to detect potential collisions, and in their initial positions, our algorithm identified a collision risk through detector 41, as shown in Fig. 25. The respective ship routes were simulated, and the outcome revealed that while the own ship exited the studied map, the target ship closed in for a collision from the starboard side, as shown in Fig.26. The algorithm's prediction was therefore confirmed as accurate. Notably, our algorithm was able to predict the collision risk within approximately 0.3 seconds, demonstrating its rapid detection capabilities. The quick detection time of our algorithm would be a crucial help for officers on board to take evasive actions, highlighting its potential practical benefits in collision risk detection scenarios.

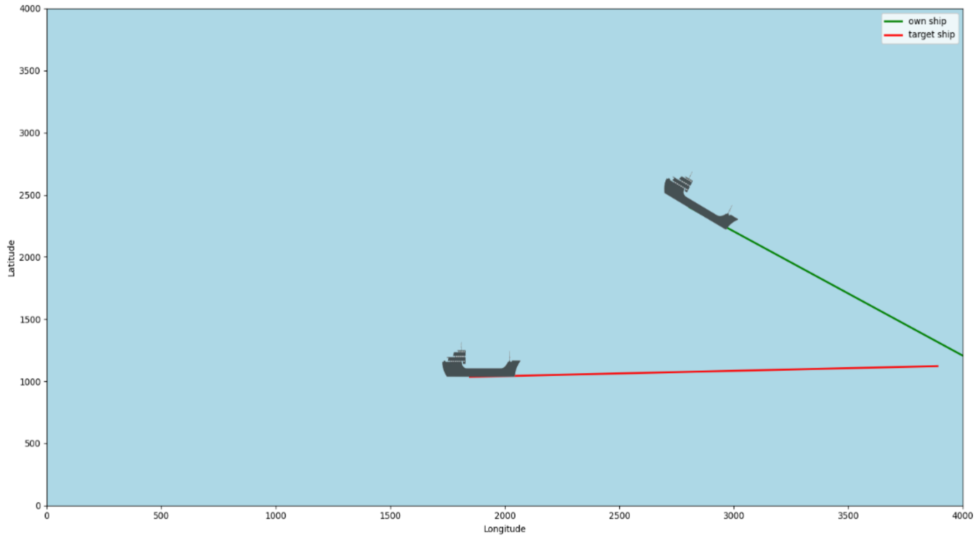

Figure 25: Case 1: Initial positions of the containerships

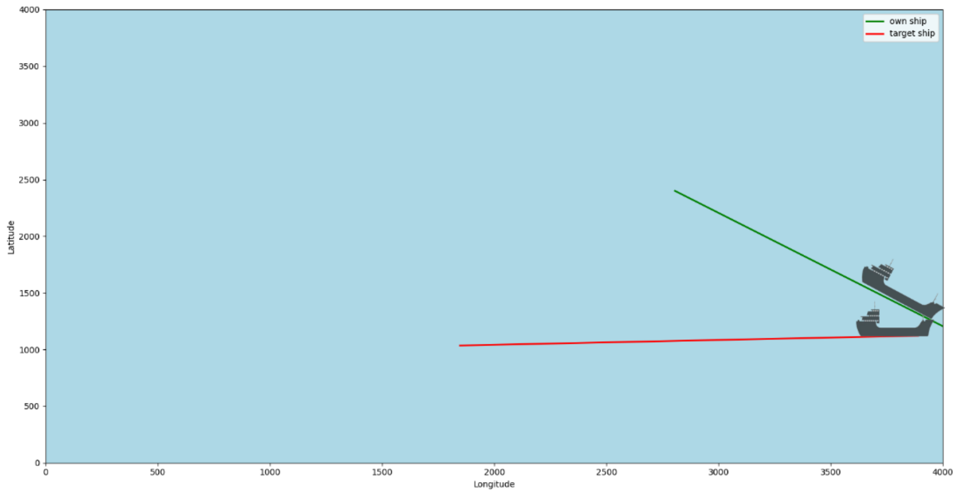

Figure 26: Case 1: Point of collision between the containerships

6.2) Case 2 – Collision

In this case a containership and a tanker with the following characteristics were generated by the script.

Table 3, Case 2: Data of the pair of ships in the initial state

	Type	Length (m)	Initial X pos	Initial Y pos	Velocity (Knots)	Heading (rads)
Own Ship	Container	350	2543	2340	14	-1.90
Target Ship	Tanker	260	3333	2007	18	0.04

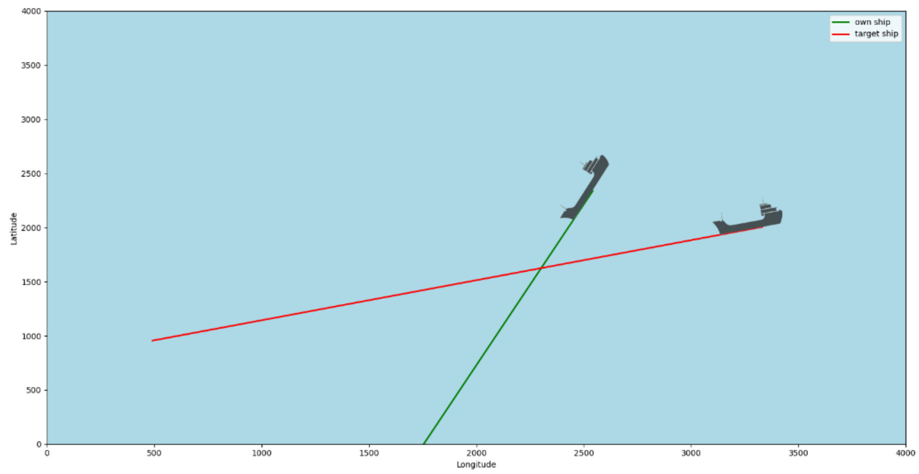

Figure 27: Case 2: Initial positions of the second pair

In this case study, we examine the scenario where the own ship is travelling southbound while the target ship travels eastward, as illustrated in Fig. 27. Our negative selection algorithm was employed to detect the collision risk using the initial positions, velocities and headings of the vessels as input. As expected, the algorithm identified that the system was operating in a non-self state, with Detector 8 covering these non-self data. The simulation results revealed a near collision incident at the point where the routes of the two ships intersect, which is demonstrated in Figure 28. This case study exemplifies the potential of our algorithm in quickly detecting a potential collision. The execution time for the algorithm was 0.15 seconds. Such swift detection time would prove to be crucial in aiding the officers on board in taking evasive actions.

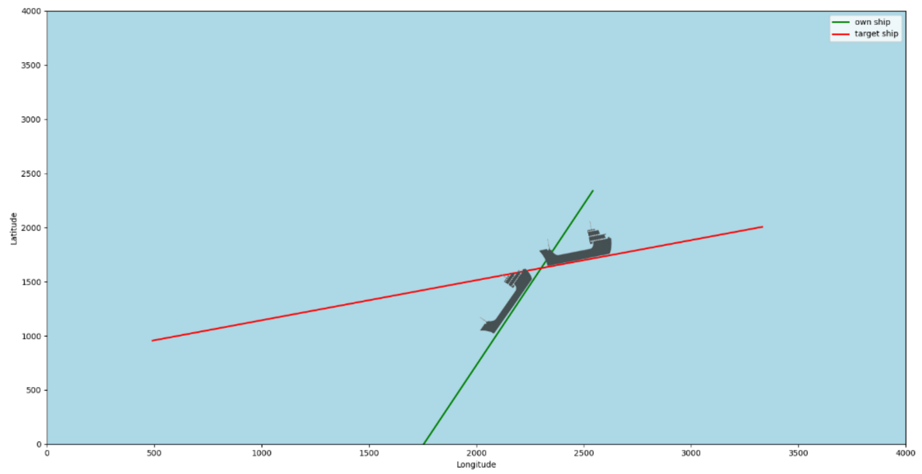

Figure 28: Case 2: Closest point of approach for the two vessels

6.3) Case 3 – Safe Navigation

In this case a containership and a fishing vessel with the following characteristics were generated by the script.

Table 4, Case 3: Data of the pair of ships in the initial state

	Type	Length (m)	Initial X pos	Initial Y pos	Velocity (Knots)	Heading (rads)
Own Ship	Container	300	426	1433	17	-1.90
Target Ship	Tanker	85	2891	959	5	-0.67

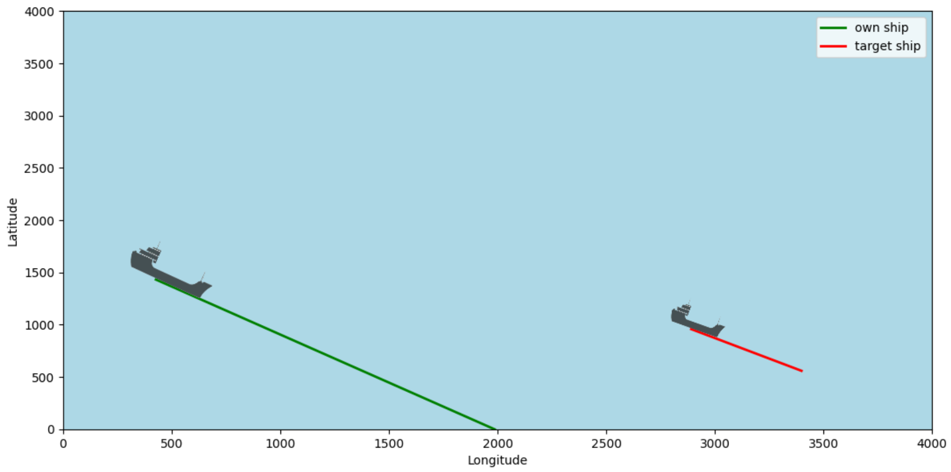

Figure 29: Case 3: Initial positions of the third pair

In this particular case study, it is evident that the initial data of the two ships do not pose any risk of collision. As demonstrated in Figure 29, both vessels are heading southbound and east, with almost parallel routes, but with the shipping vessel travelling at a much slower speed. The negative selection algorithm developed for ship collision risk detection confirms this observation, as the initial data are detected by the self hypersphere of the 13th self-sample, indicating that the data belong to the self-set. Figure 30 shows the ships when the own ship exits the map, thus further supporting the algorithm's prediction. The execution time for the algorithm to classify the initial data as self-data was approximately 0.1 seconds. The results demonstrate that the algorithm can serve as a rapid layer of safety for the officers on watch.

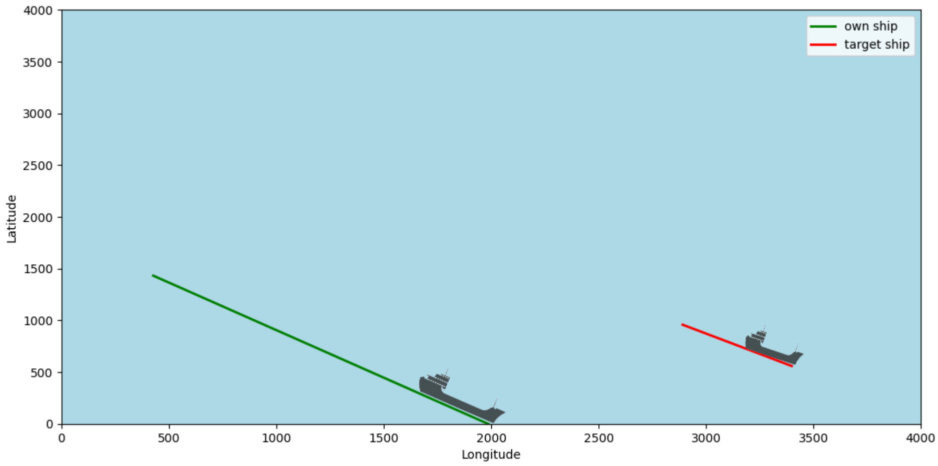

Figure 30: Case 3: Own ship exits the map

6.4) Case 4 – Safe Navigation

In this scenario a tanker and bulk carrier with the following characteristics are generated.

Table 5, Case 4: Data of the pair of ships in the initial state

	Type	Length (m)	Initial X pos	Initial Y pos	Velocity (Knots)	Heading (rads)
Own Ship	Tanker	290	2215	2416	15	-2.90
Target Ship	Bulk carrier	200	3330	438	16	1.6

In this case study, we analyze a scenario where the own ship is traveling westward while the target ship is moving northward, as depicted in Fig. 31. It is evident that there is no potential for collision between the two vessels, and our negative selection algorithm promptly confirms this observation using the initial data as input. The hypersphere generated by the 6th self sample successfully detects the initial data, characterizing it as a self state. The algorithm takes approximately 0.1 seconds to classify the data, and as illustrated in Fig. 32, the own vessel safely exits the map. This case study demonstrates the efficiency and reliability of our algorithm in providing quick and accurate safety assessments for officers on watch.

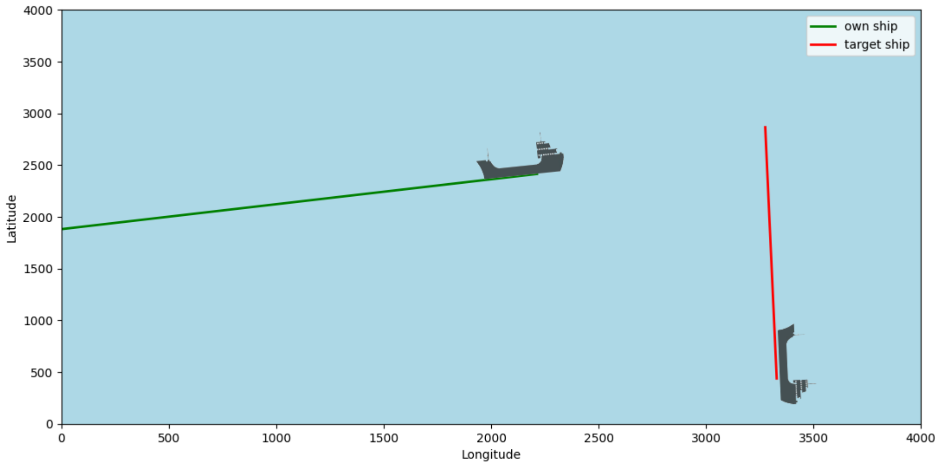

Figure 31: Case 4: Initial positions of the fourth pair

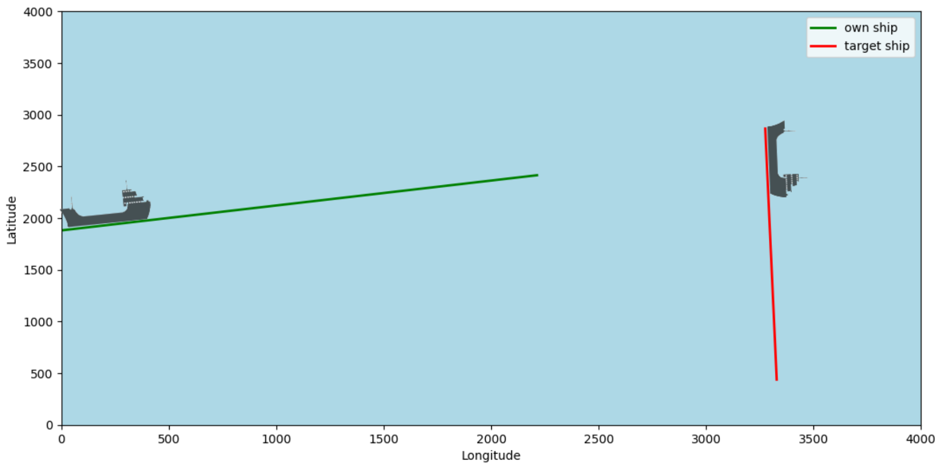

Figure 32: Case 4: Own ship safely exits the map

6.5) Conclusion

In conclusion, the presented case studies demonstrate the potential of the negative selection algorithm in ship collision risk detection. The algorithm was applied to four different scenarios, two of which resulted in a near collision, while the other two were confirmed as collision-free. The algorithm was able to detect the non-self states quickly, providing a layer of safety for the officers on watch. The execution times of the algorithm were found to be relatively short, ranging from 0.1 to 0.3 seconds, which further highlights its potential for real-time applications. Overall, these case studies illustrate the effectiveness of the negative selection algorithm in ship collision risk detection and its potential as a reliable tool for improving maritime safety.

Conclusion

Ship collision risk detection is a critical problem that has significant implications for the safety of maritime transportation. Collisions between ships can result in catastrophic consequences, including loss of life, environmental damage, and economic losses. In recent years, there has been an increase in the number of ships operating in busy waterways, leading to a higher risk of collisions. Moreover, factors such as adverse weather conditions, equipment malfunctions, and human error can further exacerbate the risk of collisions.

Given the potential consequences of ship collisions, it is essential to develop effective methods for detecting and preventing such incidents. Various methods have been proposed in the literature to address this issue, with two main categories of methods: expert-based and model-based.

On one hand expert based methods rely on human expertise and domain knowledge to identify potential collision risks. These methods often involve visual observation and analysis of radar or AIS data, and may also consider factors such as weather, vessel characteristics, and navigational rules. While expert-based methods can be effective, they are often subjective and can be prone to errors due to human factors.

In contrast, some researchers recognized the need for objective methods that would mathematically model the physics of a collision. These model-based methods can offer widely accepted solutions to the collision risk detection problem. However, a significant drawback is that in order to quantify a complex process, simplifications must be adopted to obtain a computable solution. These simplifications make the methods suitable only for restricted applications. Therefore, while model-based methods were developed to minimize human subjectivity, they are limited to cases that align with their adapted assumptions.

The method developed in this paper resides between expert-based and model-based methods, as it belongs to the field of artificial intelligence. The method considers the physics of a collision to determine which parameters could best indicate it. Once the main parameters are determined, the method uses data to detect collision or near-collision incidents. Expert knowledge can be utilized in this process, and in this thesis, such expertise was used to acquire the dataset for the algorithm's development.

The proposed algorithm is inspired by the human immune system and falls under the category of Artificial Immune Systems (AIS). Researchers in AIS have recognized the immune system as a toolbox of techniques that can be utilized to solve various complex problems. Our algorithm draws inspiration from Negative Selection Algorithms (NSAs), a subcategory of AIS. NSAs are developed based on how T-cells in the human immune system detect possibly damaging antigens while avoiding the cells of the host organism.

Similarly, NSAs are used to detect anomalies in a system by simplifying the study of a complex system to the study of a hypercube produced by its main parameters. This approach enables the algorithm to detect anomalies with greater ease, speed, and without any restrictions compared to mathematical models developed for the same purpose.

In our study, we aimed to evaluate the ability of a Negative Selection Algorithm (NSA) to detect anomalies in a system involving two ships navigating through a waterway. The system was categorized into two conditions, normal-healthy and abnormal-unhealthy, where the former indicates the safe navigation of the two ships, and the latter indicates a collision incident between them. To select the main parameters that indicate the system's health, we utilized the knowledge provided by a model-based risk detection method, aiming to combine the techniques of NSAs with marine researchers' expertise.

To train the detectors, which play the role of antibodies detecting the harmful antigens (collision incidents), our method requires data. To produce these data, we developed a secondary algorithm based on the concept of ship domain, which can be considered an expert-based method of collision risk detection. We chose this criterion due to its straightforwardness and the desire to integrate expert and model-based methods.

It is worth noting that the proposed algorithm is flexible and adaptable and therefore works effectively with various types of datasets generated by other methods, given that the identified health indicating parameters accurately capture the essential characteristics of the system under study. This capability not only broadens the applicability of the algorithm but also underscores the potential of combining expert and model-based methods to achieve a more comprehensive and robust solution to the problem of collision risk detection in ship navigation.

As with any algorithm, the performance of the Negative Selection Algorithm (NSA) in collision risk detection is heavily dependent on its main parameters. In order to evaluate the sensitivity of our algorithm to changes in these parameters, we conducted a sensitivity analysis. The analysis focused on three key parameters: the self radius, detector radius threshold, and number of detectors. Through this analysis we were also able to define which values we should adopt for our application of the algorithm. The results of this analysis are presented and discussed below.

Self radius is a crucial parameter in Negative Selection Algorithms (NSAs), as it defines the size and boundaries of the self hyperspheres, which enclose the self-samples and represent the regions where every point is also a self-sample. The self radius needs to be carefully chosen to ensure that it is large enough to capture all self-samples but small enough to avoid false positives from non-self samples. Additionally, the self hyperspheres determine the space that the mature detectors should cover as they are the first input of the algorithm. However, in scenarios where the detectors have a minimum detector radius, a large self radius can prolong the training process and lead to an inaccurate mature detector set. Through sensitivity analysis, we discovered that increasing the self radius led to a decline in the algorithm's accuracy while increasing the execution time. Thus, we chose the value of our self radius to ensure our goals of accuracy and quickness.

In our algorithm, the use of larger detectors is preferred, and therefore we have implemented a minimum detector radius as a threshold value. The detector radius is a critical parameter in our algorithm, as it determines the ability of the detectors to identify non-self cases. However, a very large detector radius can lead to decreased effectiveness in detecting non-self cases due to reduced agility in covering all areas. This was demonstrated in our sensitivity analysis, where we found that increasing the minimum detector radius initially increased the accuracy of the algorithm, but beyond a certain point, the accuracy sharply decreased and the execution time sharply increased.

The last parameter that was studied in our analysis was the number of detectors. Intuitively we can understand that if the self radius and the detector radius are determined appropriately, a larger number of detectors can better cover the non-self radius and thus achieve a higher accuracy. Nevertheless, having more detectors can greatly increase the execution time of the detector training process. As shown through our sensitivity analysis, a larger number of detectors leads to better accuracy but after a point the increase rate of the execution outperforms the increase rate in the accuracy.

The number of detectors is a critical parameter in the Negative Selection Algorithm, as it determines the coverage of the non-self space and consequently affects the accuracy of the algorithm. However, increasing the number of detectors can also lead to longer execution times during detector training, which is an important consideration in practical applications. Our sensitivity analysis revealed that increasing the number of detectors does indeed improve the accuracy of the algorithm up to a certain point, after which the marginal increase in accuracy diminishes and is outweighed by the increase in execution time. Therefore, choosing the optimal number of detectors requires a trade-off between accuracy and execution time, depending on the specific needs of the application.

Following our sensitivity analysis, we determined specific parameter values for the minimum detector radius, self radius, and number of detectors in our algorithm. To evaluate the algorithm's performance, we conducted tests on a set of non-self cases, and measured accuracy and time efficiency in a similar manner to our sensitivity analysis

In each iteration of our algorithm, we observed that non-self cases were never classified as self cases. This result holds high significance, as it implies that in a potential real-life application, the misclassification of non-self cases as self would lead to the erroneous classification of a critical situation, such as an imminent collision, as a state of safe navigation between the two ships. Thus, our algorithm's ability to accurately distinguish between self and non-self cases is a crucial aspect of collision risk detection, making it a valuable tool for enhancing maritime safety.

Another primary advantage of our algorithm is its small execution time. Despite our limited computational capacity, our algorithm can train a set of 80 mature detectors and identify a set of 30 cases within a matter of seconds. Furthermore, when provided with a ready detector set, the algorithm can identify an individual case in fractions of a second. This feature is particularly significant in a potential real-life application, as it enhances the algorithm's usability and makes it an efficient tool for real-time collision risk detection.

The algorithm's accuracy was also noteworthy, with an average accuracy of approximately 65%, and in some iterations, accuracy levels exceeding 70%. This performance demonstrates the algorithm's potential for effectively detecting collision risks between ships. With further refinement and optimization, it is plausible that the algorithm can achieve even higher levels of accuracy. The findings presented in this thesis lay the foundation for future work and underscore the importance of continued research in this area.

The algorithm's effectiveness was further demonstrated through some indicative case studies. These studies revealed the algorithm's capability to predict the outcome of a situation based solely on the information available at a specific time, without the need for simulations or advanced calculations at each time step. This result highlights the algorithm's efficacy and practicality, making it a valuable tool for real-time collision risk detection.

At this point we should mention that although our algorithm demonstrated promising results in collision risk detection, we could not compare its performance against other existing methods due to the lack of access to such algorithms. However, we acknowledge the importance of benchmarking our algorithm against other methods in the future. Such comparisons would enable a comprehensive evaluation of our algorithm's performance and its potential to enhance maritime safety.

An additional potential limitation of the algorithm is its inability to adapt to novel information. To address this issue, future iterations of the algorithm could incorporate a user feedback mechanism, which would enable the algorithm to learn from cases it fails to identify. Additionally, a feature could be implemented that enables detectors to store information regarding the specific situations they identify. These advancements would facilitate dynamic optimization of the algorithm through real-time integration of practical knowledge.

While the results of this study demonstrate the potential effectiveness of the algorithm, it is important to further evaluate its performance using real-life data. Such an evaluation would provide a more comprehensive understanding of the algorithm's potential effectiveness and limitations in real-world situations. Therefore, future work should aim to test the algorithm's performance using real-life data such as AIS Data or data taken from another validated simulation tool. Such studies would enable a more thorough evaluation of the algorithm's effectiveness and provide valuable insights for further improvements and refinement. Furthermore, these studies could provide a basis for integrating the algorithm into practical collision risk detection systems.

It is important to note that collision risk detection methods, such as the one presented in this study, can be applied to navigational situations involving more than two ships. In such cases, the algorithm can work independently with each pair of ships. This is a significant advantage of collision risk detection methods, as their primary objective is not to provide an avoidance solution, but rather to identify and assess collision risks. However, it is important to investigate whether the algorithm can maintain accuracy in multi-vessel scenarios. Therefore, potential future research would include evaluating the algorithm's effectiveness and limitations in such situations. Such studies would provide valuable insights for further improvements and practical implementation of the algorithm in real-world scenarios involving multiple vessels.

Overall, the algorithm presented in this thesis shows great potential as a collision risk detection method for maritime transportation. The sensitivity analysis and the results obtained from the case studies have demonstrated the accuracy, time effectiveness, and practicality of the algorithm. However, as with any new approach, there are still areas for improvement. The algorithm could benefit from further optimization, especially in terms of adaptability to new data and the dynamic adjustment of detectors and self points. Furthermore, the algorithm needs to be tested with real-life data to confirm its effectiveness and accuracy in actual maritime transportation scenarios. In conclusion, the results presented in this thesis provide a strong foundation for future research and development in the area of collision risk detection methods for maritime transportation.

References

1) International Maritime Organization , Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)

2) International Maritime Organization, Status of multilateral Conventions and instruments in respect of which the International Maritime Organization or its Secretary-General performs depositary or other functions, 2013

3) Allianz Global Corporate &Specialty, Shipping safety - Human error comes in many forms, 2019

4) Allianz Global Corporate &Specialty, Safety & Shipping 1912-2012. From Titanic to Costa Concordia, 2012

5) Yamin Huang, Linying Chen, Pengfei Chen, Rudy R. Negenborn, P.H.A.J.M van Gelder, Ship collision avoidance methods: State-of-the-art, 2020

6) Wikipedia, Ship Collision

7) Ship Technology, Alexander Love, Preventing ship collisions: is technology the answer?, 2020

8) Tam, C., Bucknall, R., Greig, A., Review of collision avoidance and path planning methods for ships in close range encounters, 2009

9) Lopez-Stantander, A., Lawry, J., An ordinal model of risk based on mariner's judgement, 2016

10) Phillips, Robert, 'The avoidance of accidence at sea', 1975

11) Li, X.R., Jilkov, V.P., Survey of maneuvering target tracking, 2003

12) Benjamin, Leonard, Curcio, Newman, A method for protocol based collision avoidance between autonomous marine surface craft, 2006

13) Degre, Lefevre, A collision avoidance system, 1981

14) http://www.cs.columbia.edu/~allen/F19/NOTES/cspaceexamples.pdf

15) Hvamb, K., Motion Planning Algorithms for Marine Vehicles, 2015

16) Fossen, S., Visualization of Ships in a Mixed-Reality Enviroment and Automated Situational Awareness using Live AIS Data, 2018

17) Fossen, Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles, 2002

18) Yasukawa, Yoshimura, Introduction of MMG standard method for ship maneuvering predictions, 2015

19) He, Jin, Huang , Xiong, Chen, Mou, Quantitative analysis of COLREG rules and seamanship for autonomous collision avoidance at open sea, 2017

20) Li, Liu, Negenborn, Distributed coordination for collision avoidance of multiple ships considering ship maneuverability, 2019

21) Xue, Lee, Han, Automatic collision avoidance of ships, 2009

22) Candeloro, Lekkas, Sorensen, A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels, 2017

23) Shah, Svec, Bertaska, Sinisterra, Klinger, von Ellenrieder, Gupta, Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic., 2015

24) Lefevre, Vasquez, Laugier, A survey on motion prediction and risk assessment for intelligent vehicles, 2014

25) Scheepens, van de Wetering, van Wijk, J.J., Contour based visualization of vessel movement predictions, 2014

26) Simsir, Amasyah, Bal, Celebi, Ertugrul, Decision support system for collision avoidance of vessels, 2014

27) Rong, Teixeira, Guedes Soares, Ship trajectory uncertainty prediction based on a Gaussian Process model, 2019

28) Peel, Good, A hidden Markov model approach for determining vessel activity from vessel monitoring system data, 2011

29) Chen, Huang, Zheng, Hopman, Negenborn, Cooperative multi-vessel systems in urban waterway network, 2019

30) Goerlandt, Montewka, Kuzmin, Kujala, A risk-informed ship collision alert system: framework and application, 2015

31) Kao, Lee, Chang, Ko, A fuzzy logic method for collision avoidance in vessel traffic service, 2006

32) Kuwata, Wolf, Zarzhitsky, Huntsberger, Safe maritime autonomous navigation with COLREGs, using velocity obstacles, 2014

33) Johansen, Perez, Cristofaro, Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment, 2016

34) Lee, Rhee, Development of collision avoidance system by using expert system and search algorithm, 2001

35) Chin, Debnath, Modeling perceived collision risk in port water navigation, 2009

36) Li, Pang, An approach of vessel collision risk assessment based on the D-S evidence theory, 2013

37) Zhao, Li, Shi, A real-time collision avoidance learning system for Unmanned Surface Vessels, 2016

38) Gang, Wang, Sun, Zhou, Zhang, Estimation of vessel collision risk index based on support vector machine, 2016

39) Lisowski, Game control of moving objects, 2002

40) Ozoga, Montewka, Towards a decision support system for maritime navigation on heavily trafficked basins, 2018

41) Perera, Soares, Collision risk detection and quantification in ship navigation with integrated bridge systems, 2015

42) Zhang, Goerlandt, Montewka, Kujala, A method for detecting possible near miss ship collisions from AIS data, 2015

43) Szlapczynski, Szlapczynska, Review of ship safety domains: Models and applications, 2017

44) Fuji, Tanaka, Traffic capacity, 1971

45) Coldwell, Marine traffic behavior in restricted waters, 1983

46) Goodwin, A statistical study of ship domains, 1975

47) Rawson, Rogers, Foster, Phillips, Practical application of domain analysis: port of London case study, 2014

48) Wang, Chin, An empirically-calibrated ship domain as a safety criterion for navigation in confined waters, 2016

49) Shah, Svec, Bertaska, Sinisterra, Klinger, von Ellenrieder, Gupta, Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic, 2017

50) Zhang, Villavicencio, Probability and Mechanics of Ship Collision and Grounding, 2019

51) Park, Kim, Predictive evaluation of ship collision risk using the concept of probability flow, 2016

52) Lenart, Collision threat parameters for a new radar display and plot technique, 1983

53) Huang, van Gelder, Wen, Velocity obstacle algorithms for collision prevention at sea, 2018

54) Szlapczynski, Szlapczynska, A target information display for visualizing collision avoidance manoeuvers in various visibility,2015

55) Zhao-lin, Analysis of radar PAD information and a suggestion to reshape the PAD, 1988

56) Gerhart, Larson, Shoemaker, Bruch, Ebken, Gage, Autonomous navigation and obstacle avoidance of unmanned surface vehicles, 2006

57) Fukuto, Imazu, New collision alarm algorithm using obstacle zone by target (OZT), 2013

58) Kayano, Kumagai, Effectiveness of the OZT taking into account with the Other Ship's Waypoints Information, 2017

59) Su, Chang, Cheng, Fuzzy decision on optimal collision avoidance measures for ships in vessel traffic service, 2012

60) Szlapczynski, Krata, Szlapczynska, Ship domain applied to determining distances for collision avoidance manoeuvers in give-way situations, 2018

61) Krata, Montewka, Assessment of a critical area for a give-way ship in a collision encounter, 2015

62) Baldauf, Mehdi, Fischer, Gluch, A perfect warning to avoid collisions at sea?, 2017

63) Muhammad Rozi Malim and Faridah Abdul Rahim, IMMUNOLOGY AND ARTIFICIAL IMMUNE SYSTEMS, 2012

64) Islam and Talukdar, Application of Artificial Immune System in Optimal Design of Irrigation Canal, 2020

65) N. K. Jerne, Towards a Network Theory of the Immune System, Annals of Immunology, Volume 125, 1974

66) L. N. de Castro and F.J. Von Zuben, Artificial Immune System: Part I – Vasic Theory and Applications. Technical Report 1, RT-DCA 01/99, 1999

67) L. N. de Castro, Immune, Swarm and Evolutionary Algorithms, Part I: Basic Models. Workshop on Artificial Immune Systems, Vol.3, 2002

68) Fabio Gonzalez, A Study of Artificial Immune Systems Applies to Anomaly Detection, The University of Memphis, 2003

69) J. A. White and S. M. Garrett, Improved Pattern Recognition with Artificial Clonal Selection. Artificial Immune Systems, 2003

70) S. I. Suliman, T. K. Abdul Rahman and I. Musirin, Artificial Immune-Based For Voltage Stability Prediction in Power System, 2006

71) B. K. Kihel and M. Beneyettou, Parkinson's Disease Recognition Using Artificial Immune System. Journal of Software Engineering and Applications (JSEA), 2011

72) . J. D. Farmer, N. H. Packard, and A. S. Perelson, The immune system, adaptation, and machine learning, 1986

73) L. N. de Castro and F.J. Von Zuben, aiNet: An Artificial Immune Network for Data Analysis. Data Mining: A Heuristic Approach, 2001

74) L. N. de Castro and J. Timmis, An artificial immune network for multimodal. Proceedings of the 2002 Congress on Evolutionary Computation, 2002

75) F. Campelo, F. G. Guimarães, H. Igarashi, J. A. Ramírez, and S. Noguchi. A Modified Immune Network Algorithm for Multimodal Electromagnetic Problems. ,2006

76) S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, Self−non-self discrimination in a computer. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, 1994

77) Y.M. Chen and M.L. Lee. Neural networks-based scheme for system failure detection and diagnosis. In Mathematics and Computers in Simulation (Elsevier Science),2002

78) D. Dasgupta, K. KrishnaKumar, D. Wong, and M. Berry, Negative Selection Algorithm for Aircraft Fault Detection, 2004

79) J. Balthrop, F. Esponda, S. Forrest, and M. Glickman, “Coverage and generalization in an artificial immune system,” in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), 2002

80)] D. Dagupta and F. Gonzalez, “An immunity-based technique to characterize intrusions in computer networks,”,2002

81) F. Gonzalez, D. Dasgupta, and R. Kozma. Combining negative selection and classification techniques for anomaly detection, Proceedings of the 2002 Congress on Evolutionary Computation (CEC’02), 2002

82) Chen Wen, Li Tao, Parameter analysis of negative selection algorithm, 2017

83) A. Papanikolaou, Μελέτη πλοίου- Μεθοδολογίες προμελέτης Τεύχος 1, 2009