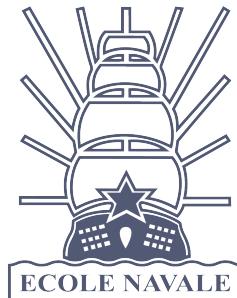


DÉTECTION TEMPS-RÉEL D'ANOMALIES CYBER SUR UN RÉSEAU NMEA PAR UTILISATION DE TECHNIQUES D'APPRENTISSAGE AUTOMATIQUE

Par

EV2 CHEVALLIER EV2 LEBIGRE

Projet de Fin d'Études



École Navale
Chaire de cyberdéfense des systèmes navals
Mai-Novembre 2020

© Copyright by EV2 CHEVALLIER EV2 LEBIGRE, 2020
All Rights Reserved

ABSTRACT

The purpose of this project is to create an algorithm based on machine learning to detect anomalies and attacks on a navigational network onboard a ship. Nowadays, all navigational equipments (GPS, Radar, Electronic Chart Display (ECDIS)...) are linked by specific network using a standard called NMEA. NMEA 0183 is a proprietary and international standard created by the National Marine Electronics Association, which presents vulnerabilities as cybersecurity was not implemented during its design process. The Global Navigation Satellite System (GNSS) is a critical part of this network and this project aims to detect anomalies in the data transmitted by the GNSS to the Electronic Chart Display (ECDIS). The detection uses machine learning techniques, as well as a visualisation plugin to alert the officer of the deck when a GPS anomaly is detected.

REMERCIEMENTS

Nous remercions le capitaine de corvette Olivier Jacq pour avoir eu l'idée et proposé ce sujet de PFE, pour son encadrement et ses conseils avisés pour ce projet, ainsi que pour nous avoir gracieusement invité à travailler dans son lieu de travail surchauffé à la Chaire de cyberdéfense des systèmes navals.

Nous remercions Clet Boudehenn, doctorant à la Chaire de cyberdéfense des systèmes navals, avec qui nous avons pu collaborer et dont l'aide fut précieuse pour la partie leurrage GPS, ainsi que Maxence Lannuzel, dont nous avons mis à l'épreuve le *plug-in* d'OpenCPN, et qui nous a aidé durant les expériences en zodiac. Merci à Madame Perrine Marziou, chargée de communication de la Chaire de cyberdéfense des systèmes navals, pour les photographies, montage et films des expériences pratiques. Nous remercions également le reste du personnel et de l'encadrement de la Chaire de cyberdéfense des systèmes navals de l'École Navale pour nous avoir donné les conditions et les ressources pour travailler.

Nous remercions le Capitaine de Corvette L'Hour et l'ensemble du Groupement d'Instruction Manoeuvre de l'Ecole Navale de nous avoir mis à disposition les moyens nautiques nécessaires à la réalisation de nos expériences pratiques.

Enfin, nous tenons également à remercier le lieutenant de vaisseau Jean Lotteau qui fut notre officier pilote durant ce PFE et le maître de conférences David Brosset, responsable de l'axe cyber et coordinateur de la chaire de cyberdéfense des systèmes navals, pour la relecture de ce manuscrit.

Table des matières

1 État de l'art	1
1.1 Les standards NMEA	1
1.2 Les vulnérabilités du standard NMEA	3
1.3 Les attaques possibles sur un réseau utilisant le standard NMEA	4
1.4 Leurrage et brouillage GPS	6
1.4.1 Le principe du positionnement par satellites	6
1.4.2 Les attaques sur le GPS	7
1.4.3 Les conséquences sur le monde maritime	8
1.5 L'apprentissage automatique	10
1.5.1 L'algorithme SVM	14
1.5.2 L'algorithme LOF	15
2 Méthodes de détection de scénarios d'attaque sur des informations GPS	17
2.1 Modélisation des effets d'une attaque sur trames GPS	17
2.1.1 Les données GPS émises sur le bus NMEA 0183	17
2.1.2 Les types d'attaque GPS : leurrage et brouillage	18
2.1.3 Détection du leurrage	20
2.2 Vecteurs d'attaque sur un réseau NMEA 0183 : attaques internes et externes	20
2.2.1 Attaque interne sur les informations GPS	20
2.2.2 Attaque externe sur les informations GPS	23

3 Expérimentations et résultats	26
3.1 Définition du problème et approche	26
3.1.1 Types d'attaques à détecter	26
3.1.2 Protocole de détection envisagé	27
3.1.3 Vérification de la pertinence d'utilisation du couple cap/distance	28
3.2 Méthodes d'apprentissage et implémentation	33
3.2.1 Méthode statistique	34
3.2.2 Apprentissage automatique avec " <i>sklearn</i> "	35
3.2.3 Score et implémentation	38
3.2.4 Affichage de la classification	41
3.3 Limites et critiques de la méthodologie de détection et des résultats	42
3.3.1 Limites de la méthode	43
3.3.2 Biais de l'apprentissage automatique	44

Liste des Figures

1.1	Exemple d'architecture d'un réseau NMEA	2
1.2	Exemple de constitution d'une trame GPS	3
1.3	Exemple de deux sommes de contrôle identiques pour des trames différentes.	4
1.4	Schéma d'une attaque par équipement piégé.	5
1.5	Triangulation par satellite	7
1.6	Le pétrolier Stena Impero arraisonné par les gardes-côtes iraniens	10
1.7	Schéma de principe du fonctionnement de l'apprentissage supervisé.	11
1.8	Recueil de l'ensemble des techniques de <i>machine learning</i> en fonction des besoins et des données.	13
1.9	Exemple de cas d' <i>overfitting</i>	14
1.10	Schéma de principe du fonctionnement de l'algorithme SVM	14
1.11	Schéma de principe du fonctionnement de l'algorithme LOF.	16
2.1	Exemple de trame GPS GGA	18
2.2	Photographie de l'écran d'un récepteur GPS victime de brouillage	19
2.3	Attaque interne du réseau par technique type " <i>man in the middle</i> "	22
2.4	Attaque externe du récepteur GPS par l'utilisation de radio logicielle.	24

2.5	Photographie prise lors d'une sortie embarcation du matériel déployé	25
3.1	Décalage de position suite à un leurrage	27
3.2	Décalage de position divergent	28
3.3	Discontinuité de distance due à un leurrage, à vitesse constante.	29
3.4	Saut de distance cap à un leurrage, à vitesse constante.	30
3.5	Zone de détection en combinant l'étude de la distance et variation de cap . .	31
3.6	Graphique de la distance entre deux points successifs en fonction de la vitesse fond sans leurrage.	32
3.7	Graphique des de la distance entre deux points successifs en fonction de la vitesse fond avec différentes valeurs de décalage à des temps différents	33
3.8	Répartition statistique de la distance entre deux points successifs normalisée	35
3.9	Essai de classification de données normales avec l'algorithme LOF.	37
3.10	Premier essai de classification de données normales avec l'algorithme SVM .	38
3.11	Exemple de circuit de test des évaluateurs avec l'embarcation.	40
3.12	Exemple de leurrage effectué par le PLUTO	41
3.13	Exemple de trames \$CY.	42
3.14	Images du rendu visuel de l'indicateur sur l'ECDIS.	42
3.15	Exemple de cas de leurrage où la position est fixée.	44
3.16	Le doctorant Clet Boudehenn manipulant le SDR-ADALM-PLUTO lors d'une sortie en mer	49
3.17	Diagramme de Gantt du Projet de Fin d'Études	50

LISTE DES ACRONYMES

- AIS : Automatic Identification System
- ARPA : Automatic Radar Plotting Aid
- ECDIS : Electronic Chart Display Information System
- GNSS : Global Navigational Satellite System
- GPS : Global Positioning System
- HDOP : Horizontal Dilution of Position
- IA : Intelligence Artificielle
- LOF : Local Outlier Factor
- NMEA : National Maritime Electronics Association
- PNT : Position Navigation Time
- RADAR : RAdio Detection And Ranging
- SVM : Support Vector Machine

Partie 1

État de l'art

1.1 Les standards NMEA

L'association américaine *National Maritime Electronics Association* (NMEA) a développé une série de standards propriétaires permettant de faciliter l'interconnexion entre les ordinateurs et autres équipements électroniques hétérogènes présents à bord d'un navire.

Parmi ces standards, celui appelé NMEA 0183 est actuellement le plus répandu à travers le monde, le nouveau standard (NMEA 2000), plus récent, plus complet en fonctionnalités et plus performant en termes de débit, n'étant pas encore généralisé en raison de la durée de vie des systèmes embarqués à bord des navires [1].

Les équipements concernés sont multiples [2], et se divisent en deux groupes : les *talkers*, émettant l'information, et les *listeners*, recevant et interprétant les trames. Parmi les nombreux équipements utilisant ce standard, nous pouvons citer :

- les systèmes de positionnement par satellites : les systèmes américains *Global Positioning System* (GPS), russe GLONASS, chinois BEIDOU et européen GALILEO,
- les indicateurs de mouvement du bateau : loch, gyrocompas et capteur d'angle de barre,

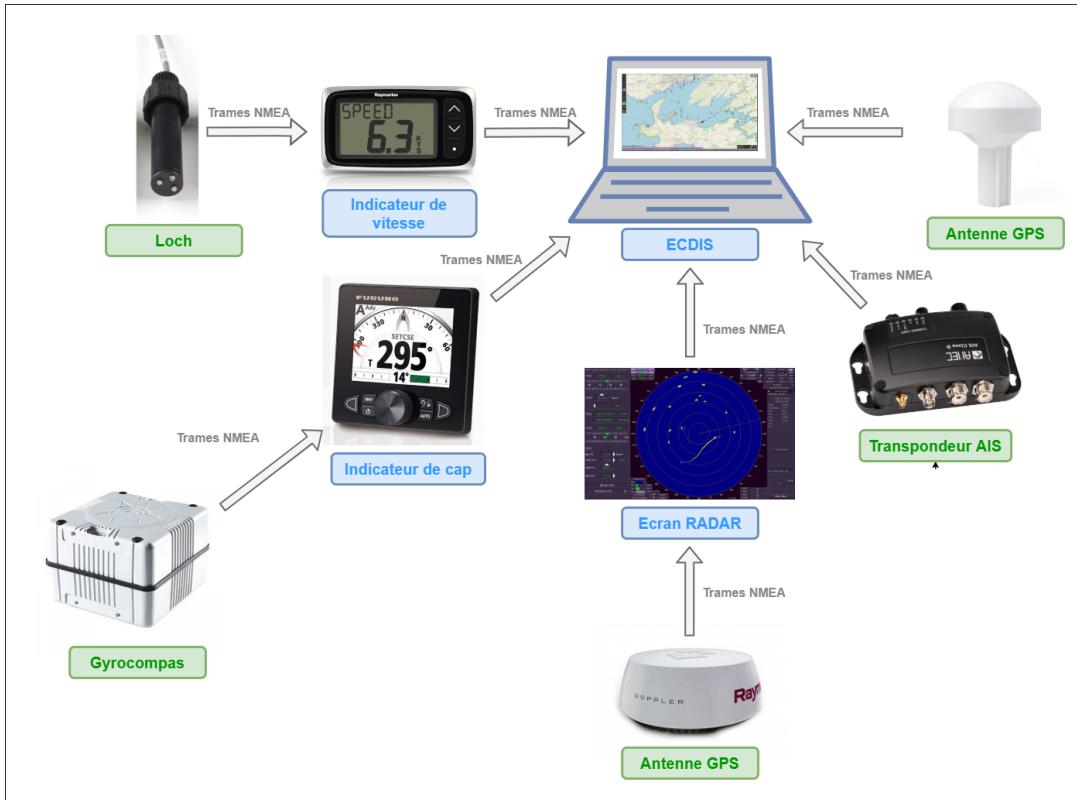


Figure 1.1 Exemple d'architecture d'un réseau NMEA

- les capteurs donnant des informations sur la situation extérieure : RADAR (RAdio Detection and Ranging) et AIS (Automatic Identification System),
- les capteurs donnant des informations sur l'environnement : sondeurs, anémomètre, thermomètre et baromètre,
- les systèmes de navigation faisant la synthèse de ces informations notamment les ECDIS (*Electronic Chart Display Information System*).

La communication entre les différents équipements est réalisée sous la forme de trames organisées en champs de caractères ASCII séparés par des virgules, le tout commençant par un \$ et terminant par un retour à la ligne. Un exemple de trame GPS est présenté à la figure 1.2 :

Une trame est définie par son émetteur, son type et ses informations spécifiques. Le

\$GPRMC,125343,A,4817.1982,N,00424.8201,W,9.50,79.65,160920,,A*53
trame , Heure , Etat , Latitude , Longitude , Vitesse , Route , Date , Checksum .

- **\$GPRMC** : Type de trame : (Il en existe de plusieurs sortes)
- **125343** : indique l'heure sous forme : hhmmss.sss : 12h53min43s
- **A** : État : A= données valides, V=données invalides
- **4817.1982** : latitude exprimée en dd.mm.ss.sss : $48^{\circ}17'19.914''$ ($^{\circ}$:degré, ' :minutes, " :secondes)
- **N** : indicateur de latitude : N= Nord, S= Sud
- **00424.8201** : longitude exprimée en dd.mm.ss.sss : $424.8201' = 424^{\circ}36.846''$ ($^{\circ}$:degré, ' :minute, " :seconde)
- **W** : indicateur de longitude : E = Est, W = Ouest
- **9.60** : Vitesse en noeud (1 noeud = 1.852km/h)
- **79.65** : Route sur le fond en degré
- **160920** : date exprimée en "ddmmaa" : 16 Septembre 2020
- **,** : déclinaison magnétique en degrés (souvent vide pour le GPS)
- **,** : sens de la déclinaison, E= Est, W= Ouest (souvent vide pour le GPS)
- **A*53** : contrôle de parité de la trame (checksum)

Figure 1.2 Exemple de constitution d'une trame GPS

premier champ, toujours de cinq caractères ASCII se décompose en deux parties : les deux premiers caractères permettent d'identifier le type d'émetteur (ici GP représente un système GPS) et les trois derniers identifient le type de phrase envoyée. Dans le vocabulaire du standard, on parle respectivement de *talker* et de *sentence ID*. Enfin, chaque trame se termine par un *checksum* (condensat), permettant de vérifier la cohérence globale via un XOR réalisé entre chaque caractère et encodé sur deux caractères hexadécimaux.

Pour chaque *sentence ID*, la fréquence des trames est définie constante selon la spécification de la norme [2].

1.2 Les vulnérabilités du standard NMEA

L'intégrité et la validité des informations transmises utilisant le standard NMEA sont d'une importance capitale pour un navire : des alertes de sécurité sont déclenchées et des effecteurs sont actionnés par le flux de trames NMEA comme, par exemple, le pilote automatique.

Pourtant, le standard NMEA, bien que largement utilisé pour permettre aux équipements marins embarqués de communiquer, présente des vulnérabilités facilement exploitable par un tiers. Dans cette partie, nous nous intéressons à différents scénarios dans lesquels un tiers utilise une faille du protocole pour influer sur le flux de données.

Le standard NMEA, qu'il s'agisse de la version 0183 ou 2000, présente deux principales vulnérabilités : tout d'abord, les informations échangées entre les différents équipements du réseau circulent en clair via un bus, l'information n'étant pas cloisonnée entre les différents capteurs. Les données émises par un des équipements sont donc susceptibles d'être interceptées et modifiées à la volée par un attaquant.

En outre, il n'existe aucun mécanisme de vérification de l'intégrité des données échangées. En effet, chaque trame est assortie simplement d'une simple somme de contrôle. Cette somme de contrôle est calculée en réalisant l'opération XOR sur les caractères ASCII successifs de la trame. Le résultat est codé sur 2 caractères hexadécimaux [3]. Cependant, ce contrôle élémentaire permet uniquement de vérifier la cohérence globale de la trame, par exemple si une des valeurs d'un champ a été modifiée. Comme il existe 256 sommes de contrôles différentes, des collisions peuvent facilement se produire.

```
$GPGLL,4916.0045,N,12311.12,W,225444,A*31  
$GPGLL,4916.45,N,12311.12,W,225444,A*31
```

Figure 1.3 Exemple de deux sommes de contrôle identiques pour des trames différentes.

1.3 Les attaques possibles sur un réseau utilisant le standard NMEA

Les vulnérabilités présentes ci-dessus peuvent être exploitées pour conduire des attaques sur un réseau utilisant le standard NMEA.

Attaque réseau

Comme le réseau n'est pas protégé par des mécanismes de chiffrement, l'information circule en clair entre les différents équipements, il est donc possible, après avoir accédé au réseau via un des points les plus faibles, d'obtenir une maîtrise totale de l'information qui y circule.

Attaque interne par équipement piégé

Une autre menace est celle des équipements piégés : des équipements modifiés sont placés en avance de phase sur le navire et sont programmés pour envoyer des trames erronées ou inattendues [4], en agissant de sorte à réaliser une attaque de type *man in the middle* (voir figure 1.4).



Figure 1.4 Schéma d'une attaque par équipement piégé.

On peut imaginer l'exemple d'un sondeur piégé qui enverrait des trames RADAR erronées ou encore qui intercepterait des trames GPS. Nous détaillons plus loin dans cette étude les effets de trames NMEA illégitimes sur la navigation

Brouillage des capteurs

Il est aussi possible d'attaquer le réseau NMEA à partir de l'extérieur, en trompant les capteurs utilisés : GPS, RADAR ou encore AIS. Ces attaques relèvent davantage de la guerre électronique mais ont pour conséquence d'affecter l'ensemble des autres éléments du réseau [5]. Il convient donc de les prendre en compte comme sources de menaces potentielles.

1.4 Leurrage et brouillage GPS

1.4.1 Le principe du positionnement par satellites

Il existe plusieurs systèmes de positionnement comportant chacun leur propre constellation de satellites : le GPS (*Global Positionning System*) appartenant au *Department Of State* américain, qui est le premier système à portée mondiale, le GLONASS russe, le BEIDOU chinois, le GALILEO européen... Le GPS étant de loin le système utilisé sur la plupart des navires et sur le matériel présent au sein de la chaire, nous avons réalisé notre projet sur ce système spécifique. Le positionnement par satellites repose sur le principe de triangulation : une constellation de satellites se trouve en orbite autour de la Terre. Chaque satellite décrit une trajectoire connue grâce à ses éphémérides, tables qui précisent la position de chaque satellite en fonction du jours et de l'heure, qui sont connues et diffusées [6].

Chaque satellite envoie à une fréquence précise des trames indiquant sa position (position relative par rapport à la Terre) et l'heure d'émission précise à la nanoseconde près. Un récepteur GPS capte les signaux des satellites visibles au dessus de son horizon et établit un calcul de distance entre le satellite et lui en mesurant le décalage temporel entre l'émission de la trame par le satellite et sa réception à bord du navire. Connaissant en outre la position relative de chaque satellite, via les données fournies par leurs éphémérides, c'est à dire des tables précisant les localisations des satellites dans le temps, le récepteur calcule alors sa position par triangulation (figure 1.5).

Certains satellites servent à recaler l'horloge du récepteur, la précision étant primordiale puisqu'une imprécision à la microseconde donne une imprécision géographique de l'ordre de la centaine de mètres. Ainsi, il faut au minimum 4 satellites visibles (3 données de position et 1 donnée de temps) pour établir un point [7].

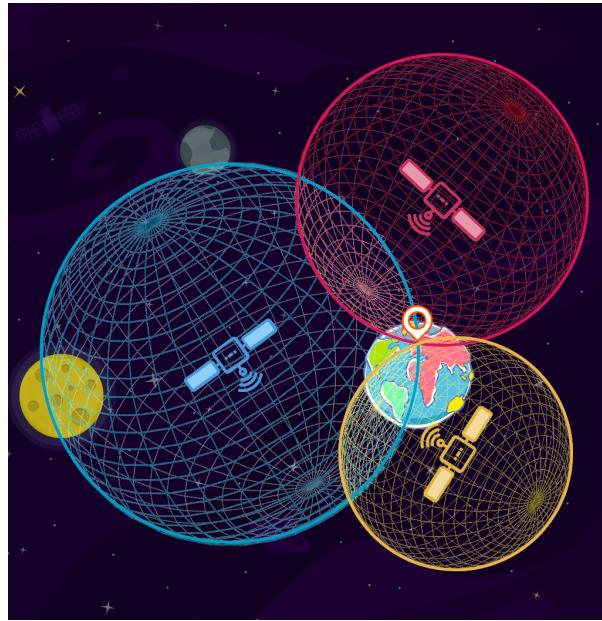


Figure 1.5 Triangulation par satellite

1.4.2 Les attaques sur le GPS

Le GPS peut être soumis à deux types d'attaques:

- le brouillage ou *jamming* qui consiste à émettre suffisamment de bruit pour empêcher la réception du signal satellite [8],
- le leurrage ou *spoofing* qui consiste à émettre de faux signaux qui se substituent aux vrais signaux satellites pour créer une nouvelle constellation. Le leurrage est une attaque plus fine et discrète plus difficile à détecter. [8]

Une combinaison des deux méthodes d'attaque est possible, en commençant par brouiller le récepteur GPS, ce qui conduit à une perte totale du signal, pour ensuite leurrer le récepteur après un apparent retour à la situation normale. Or, l'utilisation des techniques de leurrage et brouillage s'est nettement intensifiée au cours des dernières années. Ces attaques sont le ressort d'organismes étatiques et utilisées le plus souvent à des fins militaires ou politiques. D'un point de vue militaire, le leurrage permet d'entraver la liberté de manœuvre de forces

adverses en les empêchant de déterminer leurs positions sur le champ de bataille ou encore en paralysant la mise en œuvre de moyens téléguidés ou autonomes comme les missiles ou les drones. Cependant les revendications d'attaque sur les systèmes GNSS sont rares car les capacités offensives des États en terme de guerre électronique demeurent protégées par les règles du secret de la défense nationale. Ainsi, il est à l'heure actuelle difficile d'attribuer précisément et de manière fiable la responsabilité d'un incident à un acteur défini. En revanche, on dispose quand même d'une multitude de signaux faibles et d'indices dont la compilation montre une augmentation considérable d'incidents.

A titre d'exemple, l'organisation américaine C4ADS a publié en 2018 un rapport relatant les incidents sur les systèmes GNSS imputables aux forces armées russes. Se fondant sur des sources ouvertes et fermées, ce rapport permet de souligner une nette augmentation des capacités offensives russes dans ce domaine ainsi qu'un usage qui se généralise de plus en plus. [9]

1.4.3 Les conséquences sur le monde maritime

Pour assurer la sécurité nautique à bord d'un navire, le chef de quart utilise la synthèse des informations issues des différents capteurs dont il dispose. Cette synthèse est réalisée par l'ECDIS qui corrèle et rassemble les données issues par exemple du GPS, du loch, du gyro-compas, du sondeur, du RADAR et de l'AIS. Le chef de quart dispose alors d'informations concernant sa position, sa vitesse, son cap, la profondeur d'eau et la situation nautique autour de son porteur. Il prend alors les décisions relatives à la conduite du bâtiment en fonction de ces données. Une interruption de service sur le réseau NMEA peut donc avoir de graves conséquences sur la sécurité nautique et la conduite des opérations[10].

Le système de positionnement par satellites est un élément central du réseau, puisque d'autres équipements y sont couplés. Ainsi, les altérations des données issues du GPS peuvent aussi affecter les autres éléments du réseau NMEA, comme le RADAR ou l'AIS. En effet,

si l'information fournie par le GPS est perdue, le bâtiment ne connaît plus sa position en temps-réel; il peut donc se déplacer vers des eaux dangereuses (peu profondes) avec les risques d'échouement qui en découlent.

En plus de détecter les autres bâtiments situés à proximité via des émissions et réceptions d'ondes électromagnétiques, le RADAR inclut une fonctionnalité de calcul nommée ARPA (*Automatic Radar Plotting Aid*), qui détermine quand et à quelle distance un mobile passera au point le plus proche : on parle de CPA (*Closest Point of Approach*) et de TCPA (*Time of the Closest Point of Approach*). Ce calcul nécessite de connaître la vitesse et le cap du porteur, qui sont déterminés soit par le gyrocompas et le loch, soit par le GPS. Le cas échéant, une altération des données issues du GPS occasionnera une altération des informations calculées par le RADAR. Les équipements RADAR ARPA modernes peuvent aussi prendre en compte les données issues de l'AIS pour les intégrer sur la situation tactique qu'ils présentent. Le RADAR affiche alors automatiquement les informations reçues des autres mobiles situés en portée des ondes VHF comme la vitesse, le cap ou l'identification. Ces informations reçues permettent alors au chef de quart de mieux appréhender la situation nautique autour de lui. Enfin, un attaquant pourrait utiliser des techniques de leurrage GNSS pour falsifier sa position ou le type de son navire, en commençant par leurrer le GPS du bord puis en émettant à l'aide de l'AIS une position erronée, dans le but de tromper de potentiels adversaires.

Ainsi, même si l'art de la navigation exige une redondance des moyens et un recul sur l'information fournie par les capteurs, une altération du réseau NMEA peut être critique pour la navigation. [11] À titre d'exemple, en juillet 2017, le pétrolier britannique Stena Impero a été arraisonné par les gardes-côtes iraniens après être entré dans leurs eaux territoriales. Une analyse *a posteriori* confirmée par le Ministère de la défense américain révèle que le pétrolier aurait été au préalable soumis à une attaque par leurrage de son système GNSS [12]. Cet arraisionnement a entraîné d'importantes tensions entre l'Iran et le Royaume Uni. De plus, les systèmes GNSS ne servent pas uniquement à déterminer la position mais apportent aussi

la date et l'heure avec une précision importante de l'ordre de la centaine de milliardièmes de seconde [7]. Une altération de leurs données peut avoir de conséquences importantes sur l'ensemble des capteurs et éléments se synchronisant sur l'heure GPS. Ainsi, les attaques sur les systèmes GNSS peuvent avoir aussi des conséquences non seulement sur le réseau NMEA mais aussi sur les systèmes utilisant des informations de type PNT à bord du navire (*Position Navigation Time*).



Figure 1.6 Le pétrolier Stena Impero arraisonné par les gardes-côtes iraniens

1.5 L'apprentissage automatique

L'apprentissage automatique ou *machine learning* est une branche de l'intelligence artificielle. Les algorithmes de *machine learning* ont la particularité de construire un modèle à partir des données qu'ils reçoivent.

On distingue donc deux phases lors de l'utilisation du *machine learning* : une phase

d'apprentissage, où l'algorithme construit un modèle à l'aide de données d'entraînement et une phase de prédiction où l'algorithme utilise le modèle précédemment construit pour réaliser des prédictions sur d'autres données (voir figure 1.7) [13].

Il existe plusieurs catégories d'algorithmes en fonction du mode d'apprentissage qu'ils utilisent : l'apprentissage supervisé, où l'algorithme reçoit des données dites labellisées, c'est-à-dire qu'on attribue une classe à chaque donnée lors de la phase d'apprentissage et l'algorithme assigne une classe à une donnée inconnue pendant la phase de prédiction en se basant sur ce qu'il a "appris" lors de la phase d'apprentissage; d'autre part, l'apprentissage non supervisé, dans lequel l'algorithme recherche des caractéristiques particulières des données ou entre les données et construit un modèle par l'exploration des données d'entraînement. Il existe d'autres catégories d'algorithmes d'apprentissage automatique (apprentissage semi-supervisé, apprentissage par renforcement...) que nous ne détaillerons pas ici.

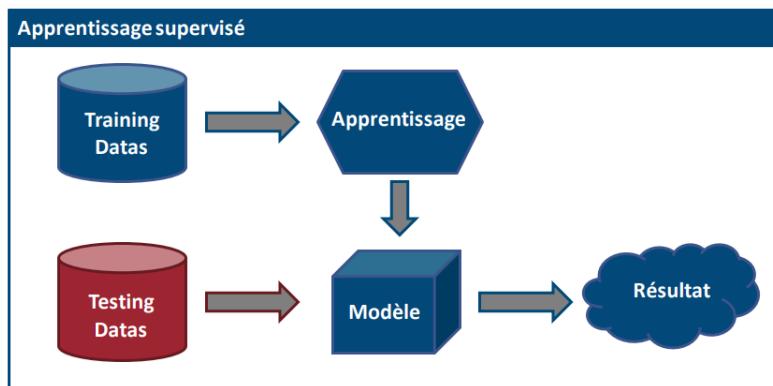


Figure 1.7 Schéma de principe du fonctionnement de l'apprentissage supervisé.

L'apprentissage automatique a de multiples champs d'application, et se décline en plusieurs catégories d'algorithmes selon l'objectif recherché (projection, détection d'anomalie, reconnaissance d'image de texte ou de langue) et du type de données. Il est ainsi possible d'utiliser ces algorithmes dans le cadre de projets visant à la reconnaissance d'images, à la détection d'intrusions et d'anomalies, ou encore la prévision de marchés financiers.[14]

Nous avons retenu le choix de l'apprentissage supervisé qui est, de manière générale, plus

adapté aux besoins de détection d'anomalies en temps réel, comme cela a pu être fait dans d'autres domaines comme la détection d'intrusions [15].

Dans le cadre de ce projet, nous cherchons à réaliser de la détection d'anomalies; pour ce faire, nous disposons de données que nous pouvons générer labellisées puisque nous réaliseras des captures de trames GPS que nous choisirons de falsifier. Ainsi, les données que nous falsifierons seront labellisées comme des données correspondant à des anomalies et les données non modifiées seront considérées comme des données correctes.

Nous commençons par déterminer un modèle à l'aide de données non modifiées, issues d'un comportement normal des équipements d'un bateau, puis ensuite nous évaluons les données à tester avec ce modèle.

Parmi un jeu de données, il est d'abord nécessaire de déterminer quels sont les paramètres pertinents à analyser : on parle également de *features* dans le vocabulaire spécifique de l'apprentissage automatique. La justification des *features*, sera détaillée plus loin.

Ensuite, en fonction des données dont on dispose, de leur quantité et type de problème auquel on est confronté, il faut déterminer quel algorithme est le plus pertinent à utiliser. Pour ce faire, comme le montre la figure 1.8, l'ensemble des techniques d'apprentissage supervisé sont répertoriées dans des diagrammes précisant quelle technique utiliser dans quel cas, éditées dans diverses publications scientifiques ou dans la documentation technique de librairies dédiées à l'apprentissage automatique. [16] Nous choisissons d'utiliser particulièrement des techniques dites de détection de la nouveauté, ou *novelty detection*. Ces méthodes seront détaillées dans la partie III.

Ensuite, il faut déterminer précisément quelles données d'entraînement seront utilisées pour obtenir le modèle recherché. Le jeu de données d'entraînement est primordial : il doit couvrir l'ensemble des situations, doit être de taille suffisante pour être représentatif, et doit correspondre à la situation à analyser sans quoi le modèle construit ne sera pas pertinent.

Par ailleurs, il faut prendre garde à entraîner le modèle de manière adaptée, car un entraînement incomplet ne permettra pas de déterminer un modèle couvrant l'ensemble des

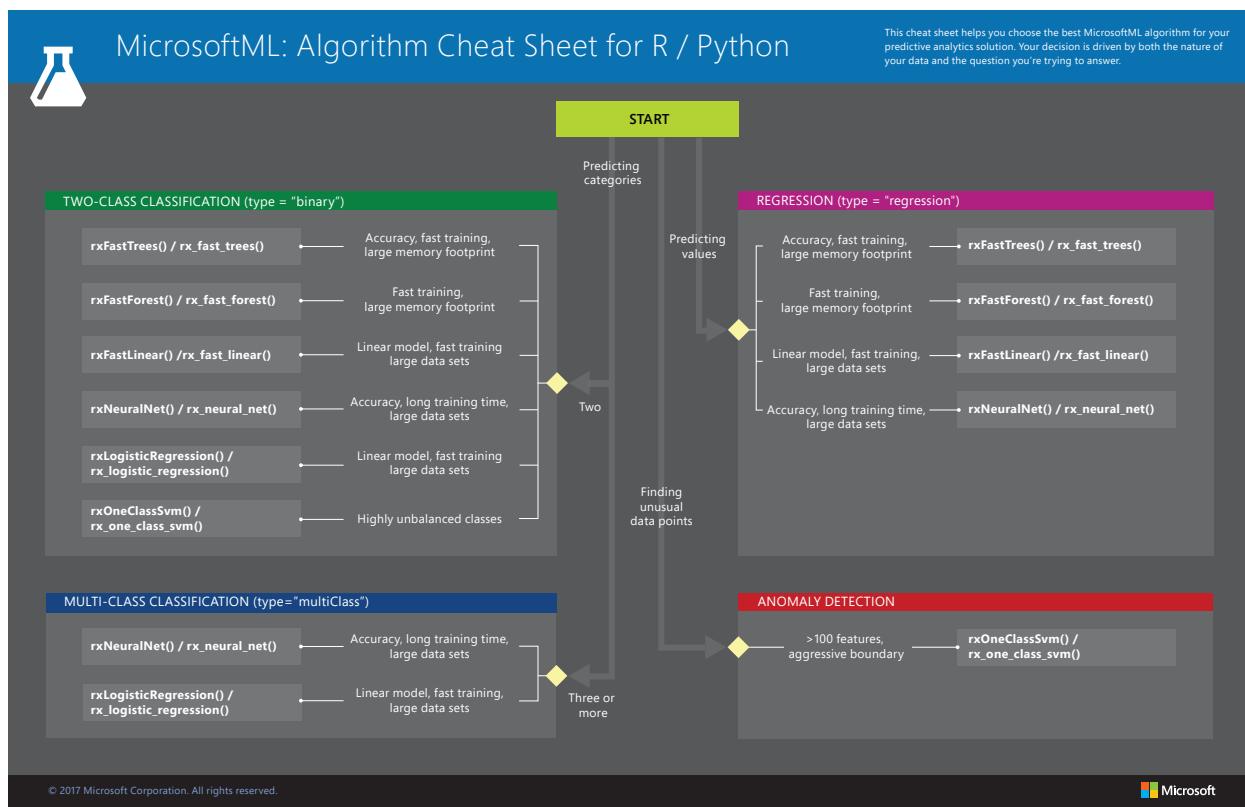


Figure 1.8 Recueil de l'ensemble des techniques de *machine learning* en fonction des besoins et des données.

situations à analyser, tandis qu'un entraînement trop intense, appelé *overfitting*, ne couvrira que les situations rencontrées sur les données d'apprentissage mais ne sera pas exploitable par la suite. Par exemple, la figure 1.9 présente deux modèles issus des points à évaluer : une régression qui est ensuite utilisable pour tester d'autres points et une interpolation polynomiale mais qu'il est impossible d'utiliser par la suite car si ce modèle correspond exactement au données d'entraînement, il n'est pas générique.

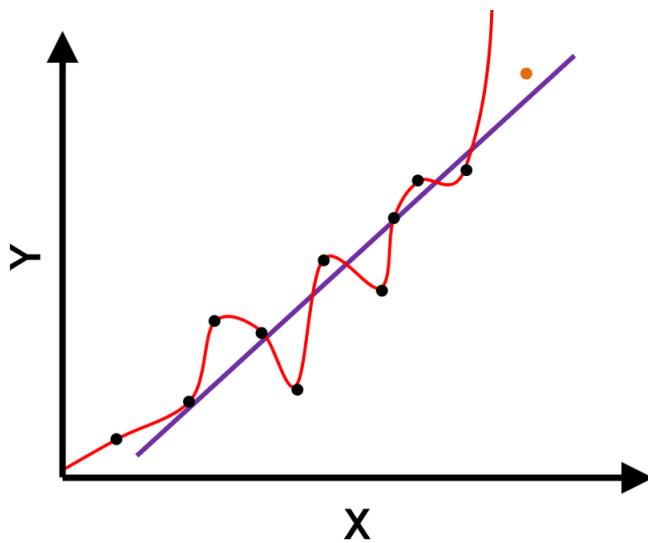


Figure 1.9 Exemple de cas d'*overfitting*.

1.5.1 L'algorithme SVM

Par la suite, pour réaliser de la détection d'anomalies, il est possible d'utiliser la méthode SVM (*Support Vector Machine*).

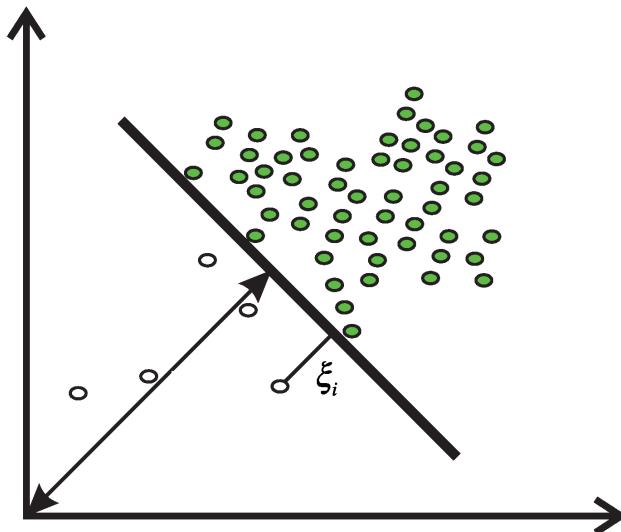


Figure 1.10 Schéma de principe du fonctionnement de l'algorithme SVM

Cette méthode consiste à déterminer parmi un jeu de données quelles sont les hyperplans qui séparent l'ensemble des points du jeu de données d'entraînement en maximisant la marge

avec l'extérieur. Une fois ces hyperplans séparateurs déterminés, il suffit de regarder avec le jeu de données de test quels sont les points situés en dehors de la zone, pour les considérer comme des anomalies.

L'intérêt de cet algorithme est qu'il est pertinent pour détecter des anomalies en temps réel (ce domaine s'appelle la détection de la nouveauté (*novelty detection*)). De plus, cet algorithme est intéressant pour résoudre des problèmes non linéairement séparables, car il permet de définir des frontières non rectilignes autour du jeu de données via un des hyperparamètres de l'algorithme. L'intérêt est dans notre cas de créer une frontière autour des points dit normaux, les points en dehors étant considérés comme anomalies, donc pouvant potentiellement provenir d'un dysfonctionnement voire d'une cyberattaque.

1.5.2 L'algorithme LOF

L'algorithme *LOF* (*Local Outlier Factor*) détermine des anomalies dans un jeu de données en calculant la densité de points dans un voisinage du point à étudier. Si cette densité est inférieure à la densité observée sur l'ensemble du jeu, on considère le point comme aberrant. [17] L'algorithme prend en paramètre le nombre de points à prendre en compte dans chaque voisinage. Comme on le voit à la figure 1.11, l'algorithme LOF permet de distinguer les points : on distingue deux amas de points et des points isolés éparpillés. Selon l'algorithme *LOF*, les amas de points sont considérés comme des données correctes tandis que les points isolés sont considérés comme des anomalies. Dans notre cas les amas de points sont les données représentant un comportement normal et les points isolés seront les données issues d'un comportement de dysfonctionnement ou d'attaque.

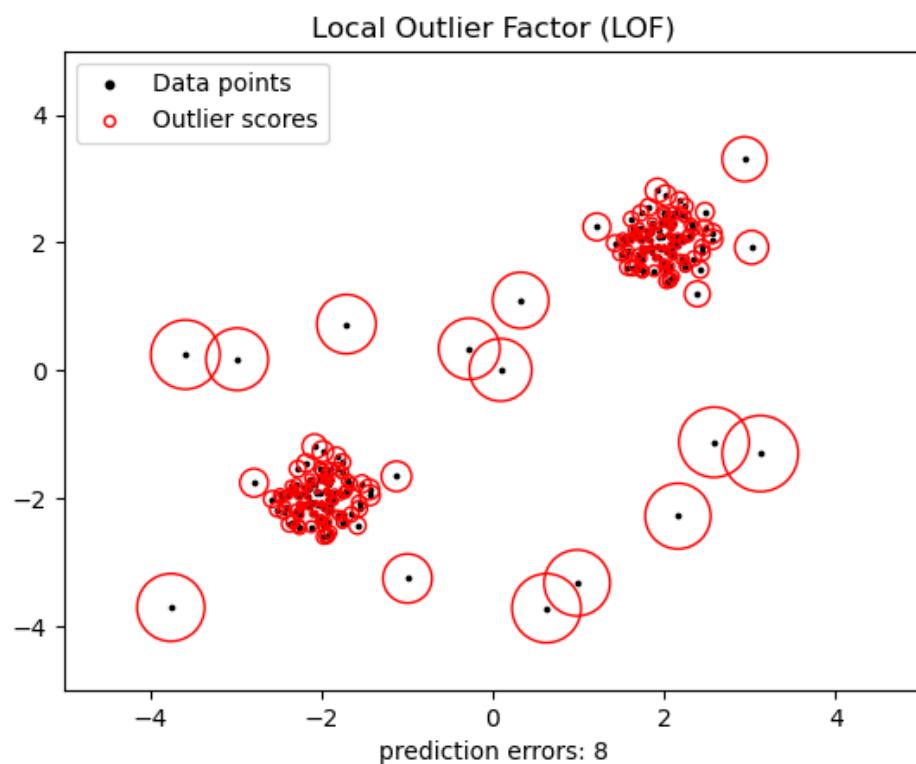


Figure 1.11 Schéma de principe du fonctionnement de l'algorithme LOF.

Partie 2

Méthodes de détection de scénarios

d'attaque sur des informations GPS

2.1 Modélisation des effets d'une attaque sur trames GPS

Les possibilités d'attaque sur les standards NMEA sont nombreuses, et peuvent concerter plusieurs types d'équipements. Dans le cadre de ce projet, nous nous intéresserons à la détection de menaces concernant le GPS, celui-ci étant aujourd'hui l'outil principal de positionnement pour les marines civiles et militaires. [7]

2.1.1 Les données GPS émises sur le bus NMEA 0183

Parmi toutes les trames émises par un GPS sur un réseau NMEA 0183, on en distingue trois principales identifiées par leur *sentence ID* :

- les trames GGL, comportant uniquement des informations de latitude, longitude et heure du point ;
- les trames GGA, comportant des informations de latitude, longitude, altitude, heure avec une précision à la milliseconde, nombre de satellites GPS visibles et le *Horizontal*

Dilution of Position (HDOP), qui est une estimation numérique de la précision du point GPS en fonction de la configuration de la constellation de satellites visibles ;

- les trames RMC, comportant la date et l'heure, la latitude, la longitude, le cap vrai, la vitesse fond et la variation magnétique.

2.1.2 Les types d'attaque GPS : leurrage et brouillage

Les conséquences des vulnérabilités du GPS, brouillage et leurrage, sont propagées sur le réseau où transitent les données NMEA. La détection d'anomalie va donc pouvoir s'appuyer sur cette propagation que nous allons mettre en avant dans ce chapitre.

Brouillage

Dans le cas d'un brouillage, le récepteur GPS ne parvient plus à discerner le signal noyé par le bruit et se retrouve incapable de calculer sa position. En effet, le signal émis par le brouilleur en champ proche est beaucoup plus fort que le signal reçu de la constellation des satellites. Au niveau NMEA, le champ indiquant la qualité du point GPS devient alors égal à zéro, ce qui indique que la position GPS n'est pas disponible. La figure 2.1 présente deux exemples de trame GGA, la première étant normale et la seconde brouillée (la figure 1.2 rappelle la description des champs). Il est intéressant de noter que, malgré le brouillage, les trames NMEA continuent à être émises avec le même rythme régulier.

```
$GPGGA,125632,4817.0062,N,00423.8243,W,1,00,2.8,3.5,M,,M,,*45
$GPGGA,125633,4817.0061,N,00423.8238,W,0,00,0,0,M,,M,,*46
```

Figure 2.1 Exemple de trame GPS GGA

Ce type d'attaque est rapidement visible. En effet, les systèmes de visualisation comme les ECDIS détectent rapidement que l'information GPS est manquante et des alertes existent

déjà pour prévenir le navigateur du manque d'information. Par exemple, le système de réception utilisé lors de nos expériences affiche une alerte lors de la perte d'information GPS (figure 2.2).

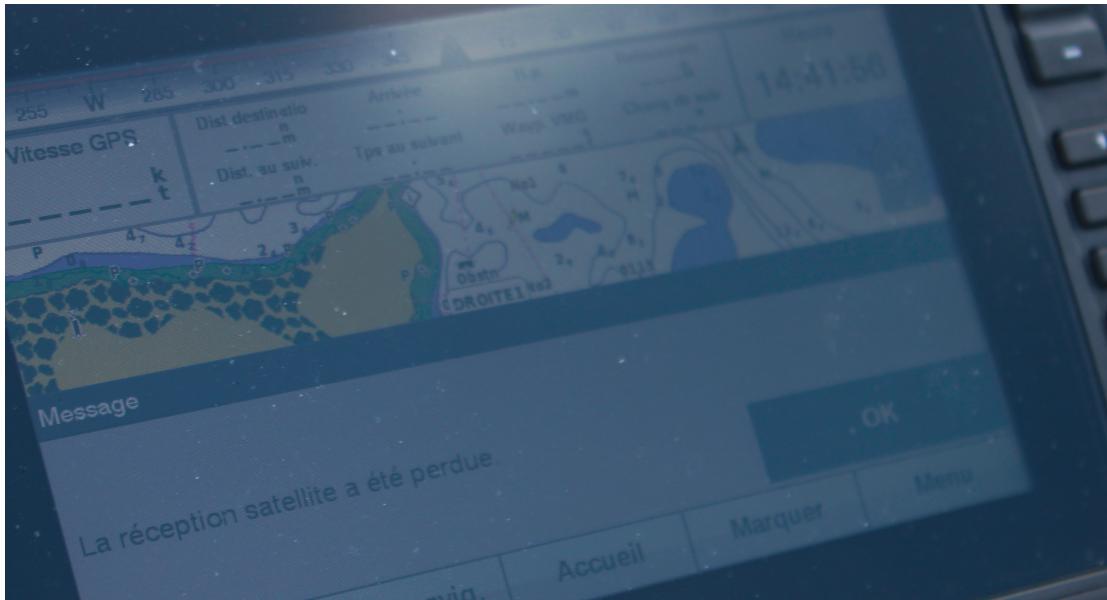


Figure 2.2 Photographie de l'écran d'un récepteur GPS victime de brouillage

Pour cette raison, nous avons fait le choix de ne pas nous intéresser à ce type d'attaque, leur détection étant assez aisée. Cependant, il est important de souligner que l'effet de perte totale d'information GPS peut avoir différentes causes : un brouillage ou peut-être seulement un dysfonctionnement du capteur.

Leurrage

Dans le cas d'un leurrage, les effets sur les trames sont bien moins visibles. L'information GPS est simplement modifiée, de manière plus ou moins subtile. C'est en raison de la difficulté de détection de ce type d'attaque, de leur augmentation, de leurs impacts sur la sécurité de la navigation et de l'absence d'outil généralisé pour leur détection que nous avons choisi de nous y intéresser tout particulièrement.

2.1.3 Détection du leurrage

Dans le cas d'un leurrage, les informations contenues dans les trames GPS sont modifiées. L'attaque qui vient spontanément à l'idée est une modification des coordonnées GPS, c'est-à-dire les deux premiers champs des trames. Cela affecte alors les autres outils de navigation et notamment l'ECDIS : le bâtiment apparaît alors à une position fausse. Cela peut avoir de graves conséquences pour le bâtiment si celui-ci utilise la position GPS comme principal système de positionnement comme évoqué en partie I, en cas de mauvaise condition météo ou de l'absence de repère côtier. Par la suite, nous nous restreindrons dans notre étude uniquement à une attaque modifiant les valeurs de latitude et longitude. Cependant, il convient de rappeler que les attaques de leurrage peuvent également affecter d'autres champs, comme le temps GPS, utilisé par de nombreux autres équipements à bord.

2.2 Vecteurs d'attaque sur un réseau NMEA 0183 : attaques internes et externes

Notre objectif est donc de détecter les cas de leurrage de position. Pour cela, il importe d'abord de reproduire les conditions d'une attaque, pour identifier les moyens de détection et développer une réponse appropriée. Les attaques ciblant le GPS peuvent avoir des vecteurs différents. Elles peuvent être d'être d'origine interne, par équipement piégé, ou externe, par leurrage via un équipement radio-électrique. Nous allons, dans la suite de cette étude, détailler ces attaques et proposer des scénarios pour chaque type, pour tenter de les reproduire et trouver des solutions pour les détecter.

2.2.1 Attaque interne sur les informations GPS

On considère d'abord une attaque en interne sur les trames NMEA 0183, c'est-à-dire via un équipement piégé qui serait présent à bord et relié préalablement au réseau NMEA.

Dans ce scénario, un équipement intercepte les trames émises par le GPS, les modifie, et les renvoie vers les récepteurs utilisant les informations issues du GPS (ECDIS, RADAR...). L'équipement d'attaque se trouve alors dans une position de type *man-in-the-middle*. L'intérêt d'une telle attaque est son faible coût puisqu'il n'y a pas besoin d'équipements sophistiqués de leurrage, un simple composant informatique embarqué de petite taille ou un composant logiciel malveillant étant suffisants. De plus, les navires devenant de plus en plus complexes et embarquant de plus en plus de matériel électronique, il est maintenant plus facile d'y introduire un système miniature et faire en sorte qu'il ne soit pas détecté et ce d'autant plus que nombre d'équipements électroniques complexes sont présentés comme des "boîtes noires" dont l'équipage ne maîtrise pas le fonctionnement. En revanche, il est nécessaire d'obtenir en amont un accès physique à la plate-forme que l'on veut attaquer et donc, potentiellement, corrompre un membre d'équipage ou un maintenancier.

Afin de réaliser ce scénario et obtenir les données afférentes, nous avons utilisé les logiciels de simulation "*Bridge Command*" [18] et "*Open CPN*" [19], accompagnés d'un script Python. Cette configuration est schématisée en figure 2.3. "*Bridge Command*" est un logiciel de simulation de passerelle, qui présente l'intérêt d'agir comme source NMEA et émettre ces données sur un réseau. Le logiciel reproduit assez fidèlement le comportement d'un navire (changement d'allure, de cap...), même si des approximations sont faites (seul certains types de trames sont produits, la montée d'allure en machine est instantanée, ce qui est différent de la réalité...). "*Open CPN*" est un logiciel libre permettant de réaliser des fonctions de type ECS (*Electronic Chart System*). Il offre ainsi la possibilité d'afficher en temps réel, sur des cartes électroniques de navigation, la position d'un bâtiment à partir des trames NMEA émises par le GPS. Ces deux logiciels communiquent via le protocole UDP (*User Datagram Protocol*). Enfin, Un script python *man in the middle* récupère les trames émises par "*Bridge Command*" et les modifie à la volée avant de les renvoyer vers "*Open CPN*". Il utilise la librairie "*Pynmea2*" pour séparer (ou *parse*) chaque champs des trames NMEA[20]. Il peut modifier n'importe quel champ de n'importe quel type de trame, ce qui simule une attaque.

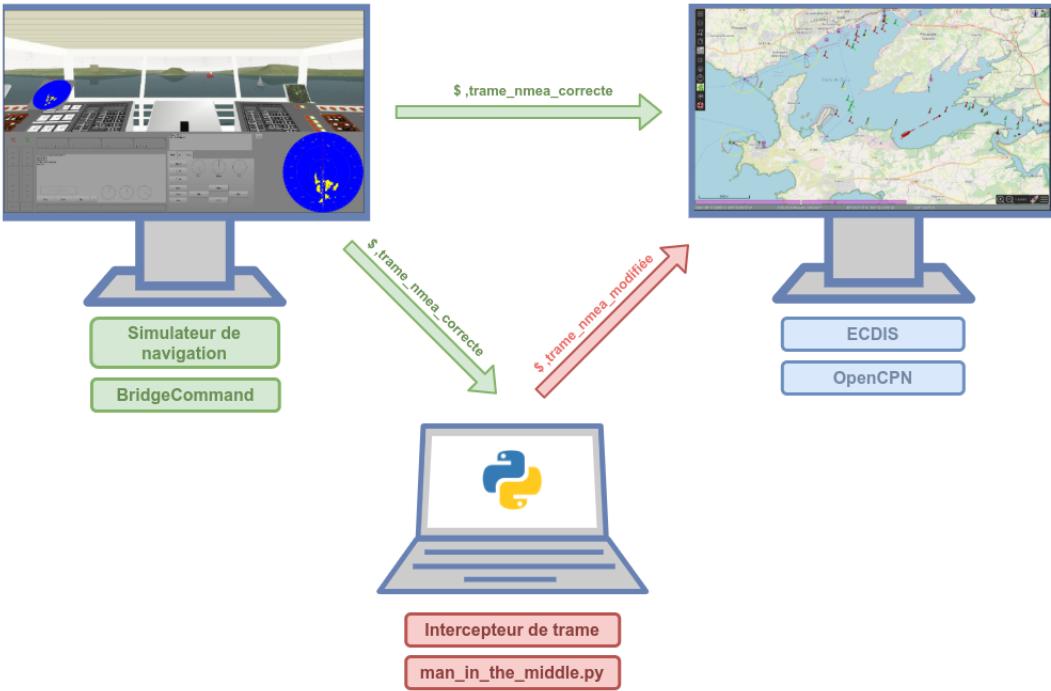


Figure 2.3 Attaque interne du réseau par technique type "*man in the middle*"

En désactivant le script Python, nous avons d'abord utilisé les deux logiciels pour générer des données légitimes (non modifiées), notamment afin de construire une base d'apprentissage pour l'apprentissage automatique. Nous avons pour cela effectué une capture longue des trames (45 minutes), en essayant de reproduire le plus de situations possibles (girations à vitesses différentes, montées et descentes en allure...). Pour obtenir un modèle plus robuste, nous aurions pu utiliser un volume de données plus importantes (il existe des captures de données NMEA de plusieurs jours), mais nous avons préféré créer notre propre jeu de données d'entraînement. Dans notre cas, cela a permis de mettre à l'épreuve nos programmes directement grâce à "*Bridge Command*" et de pouvoir reproduire plus de situations de navigation (les jeux de données disponibles sont souvent ceux de bateaux avançant en ligne droite pendant de longues périodes, et ne changeant pas de vitesse [21]). De plus, nous effectuons des tests qui ne dépassent pas quelques minutes : une durée de 45 minutes pour les données d'entraînement est donc largement supérieure à la durée de test. Par la suite, nous avons

fait fonctionner en même temps un scénario s'approchant du comportement normal d'un navire sur "*Bridge Command*", le script python modifiant en temps réel une certaine proportion des trames, et le programme de détection, afin de tester ce dernier. Cela constitue notre premier moyen de tester nos algorithmes de manière simple, mais pouvant s'éloigner de la réalité, "*Bridge Command*" ne simulant que de manière approximative la production de trames NMEA et la cinématique d'un bâtiment.

2.2.2 Attaque externe sur les informations GPS

L'équipement GPS peut aussi subir une attaque externe, par leurrage ou brouillage GPS. Dans ce cas, un équipement radio-électrique émet sur la bande de fréquences utilisée par les satellites GPS. La force des signaux reçus à terre étant particulièrement faible et les récepteurs très sensibles, la force du signal des informations leurrées dépasse celui envoyé par les satellites. Bien que ces techniques soient illégales, ce type d'équipement est aujourd'hui facile à acquérir sur Internet et largement employé par certaines marines militaires dans certaines régions du monde. [22]

Si le niveau de bruit est supérieur au signal, le signal satellite est noyé dans le bruit et le récepteur GPS devient incapable de calculer sa position, c'est alors un cas de brouillage, on observe alors une perte totale du signal. Dans ce cas, il n'y a aucune volonté de reproduire un signal GPS réaliste mais uniquement un déni d'accès au service. Dans le cas du leurrage, l'équipement radioélectrique émet un signal identique à celui que recevrait la victime si elle se situait à une position donnée. Pour cela, le système attaquant simule la constellation visible à une position donnée, à une heure donnée en se basant sur les éphémérides. Le récepteur GPS reçoit donc une information fausse qui indique une position du porteur qui ne correspond pas à la position réelle, c'est un leurrage. Le leurrage peut être grossier (avec des centaines ou des milliers de kilomètres d'écart de position), ou beaucoup plus subtil et visant un navire particulier, avec un décalage potentiellement beaucoup plus petit.

Pour réaliser ce type d'attaque en laboratoire, nous utilisons un module radio-logiciel (*Software Defined Radio*) appelé ADALM-PLUTO SDR, disponible sur le marché pour moins de 200 Euros. Cette carte électronique permet de générer des signaux dans la bande de fréquence 325-3800 MHz [23] et, une fois configuré, peut émettre des signaux semblables aux signaux satellitaires utilisés pour la navigation par satellite. Le service américain de positionnement par satellite pour l'usage civil utilise deux fréquences : la L1 centrée sur la fréquence 1575.42 MHz, et la L2 centrée sur 1227.60 MHz [25]. Ces fréquences rentrant dans la bande couverte par le PLUTO, il est ainsi possible de générer des signaux GNSS falsifiés.

La puissance d'émission du PLUTO en laboratoire (de l'ordre de 0,01 W) est bien supérieure à la puissance des signaux satellitaires (de l'ordre du pW) : le récepteur GPS utilise alors le signal falsifié et se retrouve leurré. Il est nécessaire de préciser ici que toutes les mesures ont été prises en laboratoire pour que notre expérimentation n'ait pas d'impact sur le réseau opérationnel GPS : la sphère d'émission que nous avons pu mesurer était inférieure à 1 mètre et notre puissance d'émission minimale.

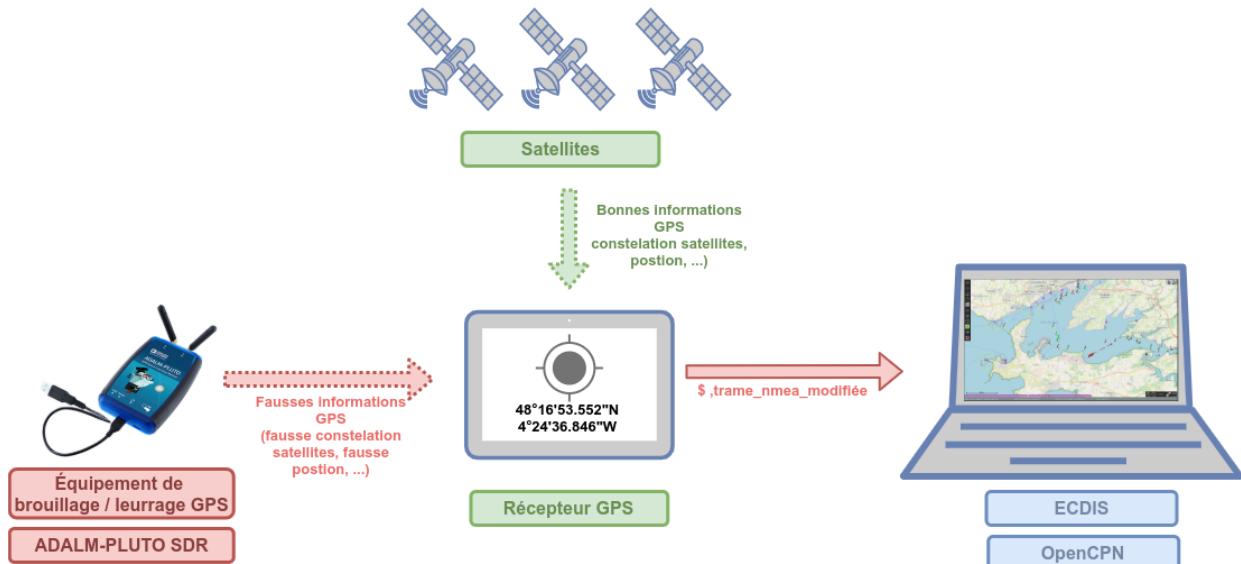


Figure 2.4 Attaque externe du récepteur GPS par l'utilisation de radio logicielle.

Pour reproduire des situations de leurrage, nous avons eu la chance de pouvoir utiliser des moyens nautiques (embarcation pneumatique) de l'École Navale pour réaliser un scénario similaire. Un récepteur GPS est monté sur l'embarcation et transmet les trames qu'il produit vers un ordinateur portable, lequel affiche la position GPS qu'il reçoit grâce à "*Open CPN*" et peut analyser les trames l'aide d'un programme de détection. De plus, un module ADALM-PLUTO SDR, présent lui aussi sur l'embarcation, peut être programmé pour brouiller ou leurrer le récepteur GPS. L'objectif est de vérifier que le programme est apte à détecter les attaques créées par l'émetteur SDR. Ce scénario est schématisé sur la figure 2.4, et la photographie en figure 2.5 montre le montage réel sur l'embarcation.

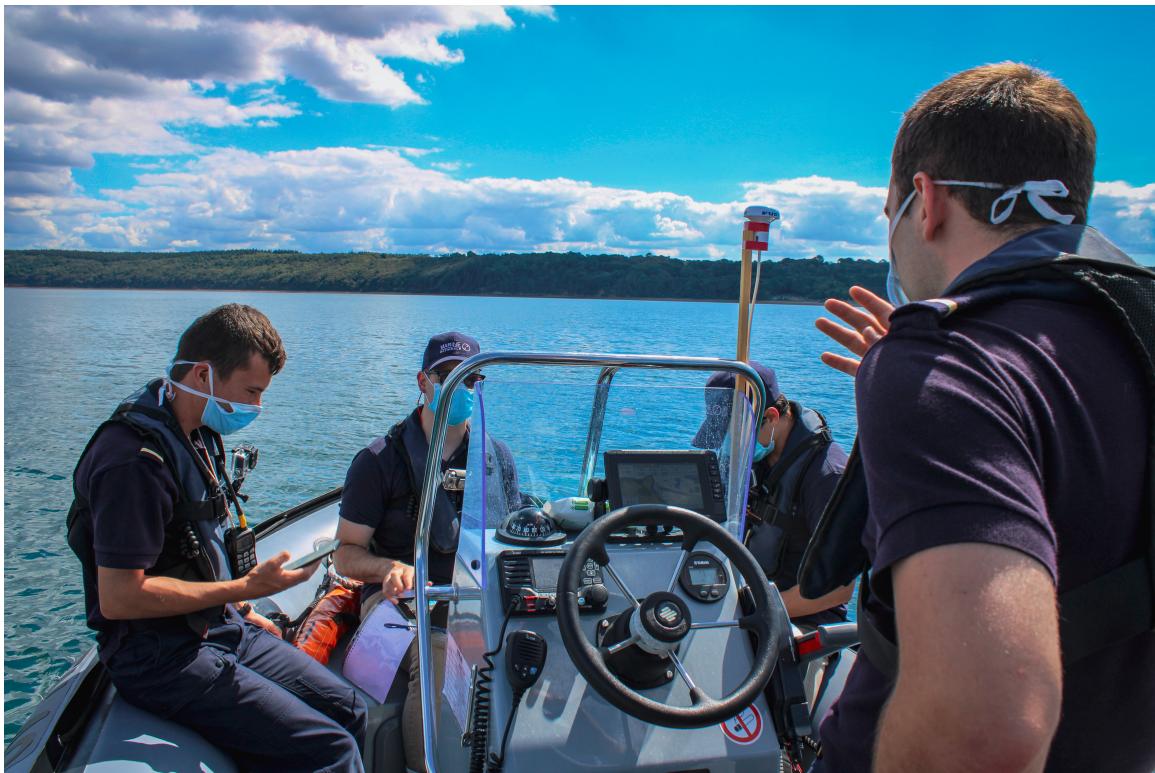


Figure 2.5 Photographie prise lors d'une sortie embarcation du matériel déployé

Notre objectif est alors de réaliser le programme analysant les trames NMEA émises par le GPS vers les consommateurs de ce type de données, comme un ECDIS, afin de détecter les attaques de ces deux scénarios.

Partie 3

Expérimentations et résultats

3.1 Définition du problème et approche

3.1.1 Types d'attaques à détecter

Les scénariis étant posés, nous nous sommes maintenant intéressés à l'algorithme de détection en lui-même. Nous faisons le choix de construire un programme pouvant détecter des attaques sur le GPS, en n'ayant accès qu'au contenu des trames NMEA. Nous considérons de plus que les attaques sont des leurrages (nous avons vu que le brouillage est déjà détecté par les appareils), et qu'elles ne concernent que les champs de latitude et longitude, la grande majorité des cas d'attaques répertoriés faisant référence à des leurrages de position (voir partie I). Enfin, nous considérons la fréquence d'émission des trames constante, et assez élevée pour considérer la vitesse du bâtiment constante entre ces deux points (la fréquence habituellement de quelques Hz pour les trames GPS) [2].

Une manière de détecter une modification de la seule position GPS serait de vérifier la cohérence de la suite des positions GPS avec d'autres données issues de trames, si possible provenant d'appareils indépendants du GPS et donc non impactés par le leurrage (par exemple : le loch). En croisant les données, il serait possible de vérifier si celles-ci sont cohérentes

entre elles.

Cependant, n'ayant accès qu'aux données issues du GPS pour construire notre modèle, nous avons opté pour des algorithmes utilisant uniquement les données GPS.

3.1.2 Protocole de détection envisagé

Une modification de la position GPS se traduit par une ou des variations inattendues de position entre deux trames successives, que l'on appellera décalage. A partir d'un instant t où le leurrage commence, le point GPS reçu est décalé, ce qui a une influence sur la différence de latitude et longitude entre le point à $t-1$ et le point à t . (voir schéma 3.1).

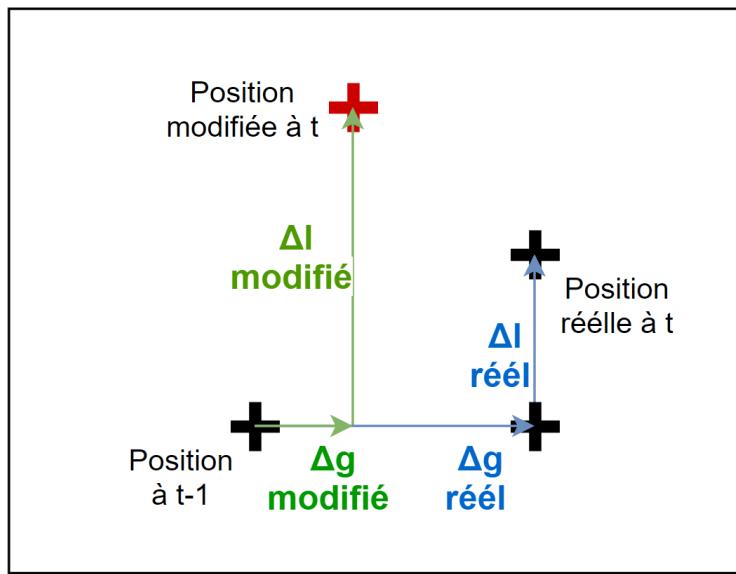


Figure 3.1 Décalage de position suite à un leurrage

Une divergence de route entre les points réels et modifiés pourra être interprétée comme une suite de décalages successifs (voir figure 3.2).

Le point clé de notre étude est de détecter ces décalages en croisant les informations de positions avec les autres champs. L'utilisation directe des champs de position ne nous paraît pas adaptée, un décalage entraînant seulement le déplacement de la position GPS vers une autre position GPS. Cette nouvelle position GPS, en supposant qu'elle se situe

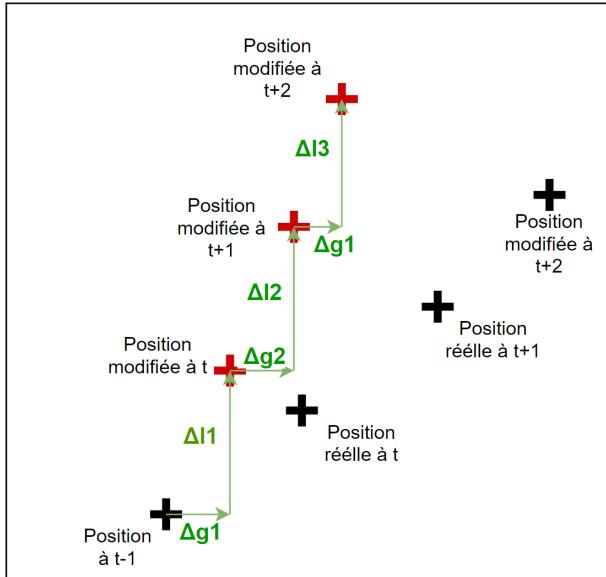


Figure 3.2 Décalage de position divergent

sur l'eau, est tout à fait atteignable par le bâtiment en théorie, ce qui rendra la détection compliquée. Nous cherchons donc à utiliser les données de position de façon différente, pour créer de nouvelles variables qui, par leurs seules valeurs et les autres paramètres étant fixés par ailleurs, permettraient de détecter un leurrage.

Pour cela, il est possible, par le calcul, de déterminer entre deux points successifs, le cap et la distance parcourue par le navire. Nous faisons alors l'hypothèse qu'un leurrage cause une discontinuité inattendue de distance ou de cap, qu'il sera possible de détecter. En effet, en situation normale, nous supposons que la suite distance et la variation de cap est bornée, et qu'une attaque peut entraîner des anomalies dans cette suite.

3.1.3 Vérification de la pertinence d'utilisation du couple cap/distance

Nous partons de l'hypothèse que, à une vitesse donnée, la distance entre deux points GPS consécutifs peut servir pour détecter une attaque : une distance trop élevée à une vitesse donnée peut être révélatrice d'un leurrage. De même, une variation de cap trop importante par rapport aux situations habituelles pourrait permettre de déceler des attaques.

D'abord, notons que la distance entre deux points GPS successifs et la vitesse fond moyenne devraient être corrélées, la distance parcourue étant en théorie proportionnelle à la vitesse. Si ce n'est pas le cas, les deux points n'ont pas une distance cohérente avec la vitesse et un décalage de position est peut-être présent. Ce dernier entraînerait une distance calculée trop grande par rapport à la vitesse (*cf* figure 3.3). Dans cet exemple, la distance calculée 2 est trop grande par rapport à la vitesse, considérée constante dans l'exemple. Il convient cependant de vérifier que la vitesse fond utilisée n'est pas calculée à partir des points déjà modifiés, sinon, aucune incohérence de distance ne pourra être détectée. C'est le cas dans les attaques que nous avons choisi de simuler (seuls les champs de latitude et longitude sont modifiés, la vitesse fond ne change pas).

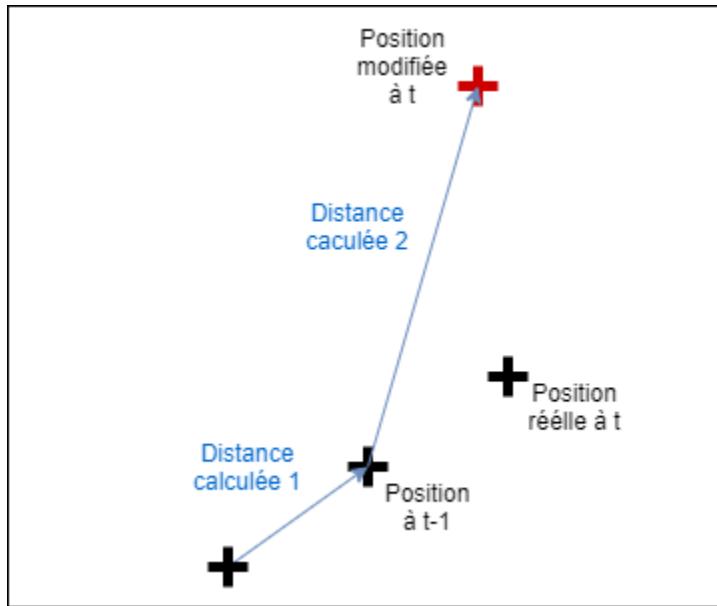


Figure 3.3 Discontinuité de distance due à un leurrage, à vitesse constante.

Dans la même démarche, il est aussi possible de comparer les champs de position et de cap : à partir d'un calcul simple, en considérant la route entre deux points constante (voir hypothèses précédentes), il est possible de déduire de deux positions successives le cap emprunté. Si la variation de cap n'est évidemment pas proportionnelle à la vitesse, on peut s'attendre qu'à une vitesse donnée, la variation de cap soit bornée (la fréquence des trames

étant de quelque Hz, l'inertie du bateau fait qu'il ne peut pas tourner à une vitesse angulaire trop grande entre deux trames, *cf* figure 3.4). Dans ce cas, la différence de cap calculée est la différence entre les caps calculés 1 et 2, égale à 45 degrés sur l'exemple. Une modification de la position pourrait alors entraîner une variation trop importante de cap entre deux positions, ce qui pourrait permettre de la détecter.

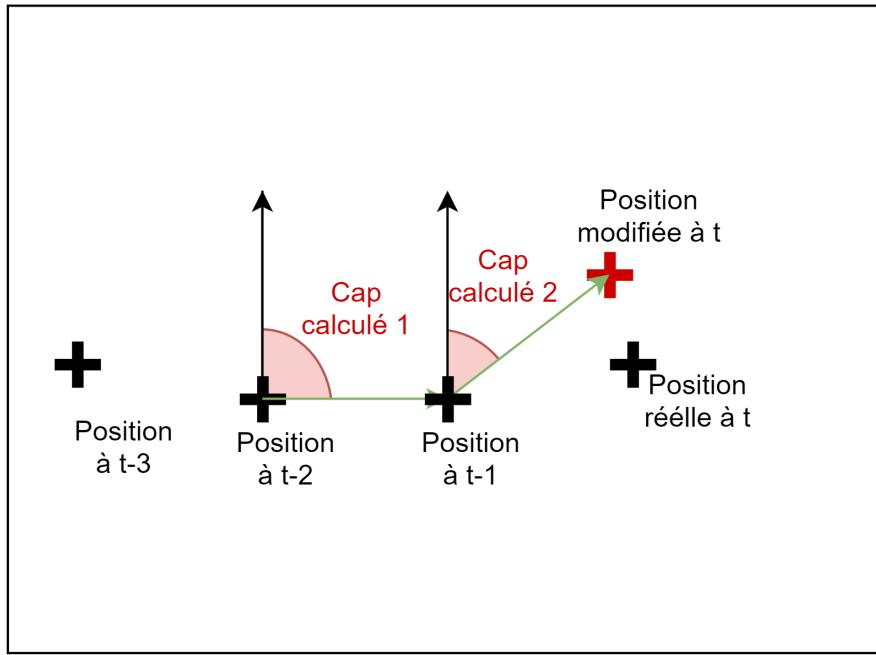


Figure 3.4 Saut de distance cap à un leurrage, à vitesse constante.

Combiner l'étude du cap et de la distance permet alors de construire une zone de détection, dans laquelle des points GPS seraient détectés comme anomalies. La méthode cap considère un point comme anomalie lorsqu'il sort d'un cône à l'avant du navire (variation de cap trop importante), alors que la méthode distance le considère lorsqu'il sort d'un cercle autour du navire, comme représenté sur la figure 3.5. Ce cône et ce cercle ont une ouverture angulaire et un diamètre qui dépendent de la vitesse et de la résolution de la méthode.

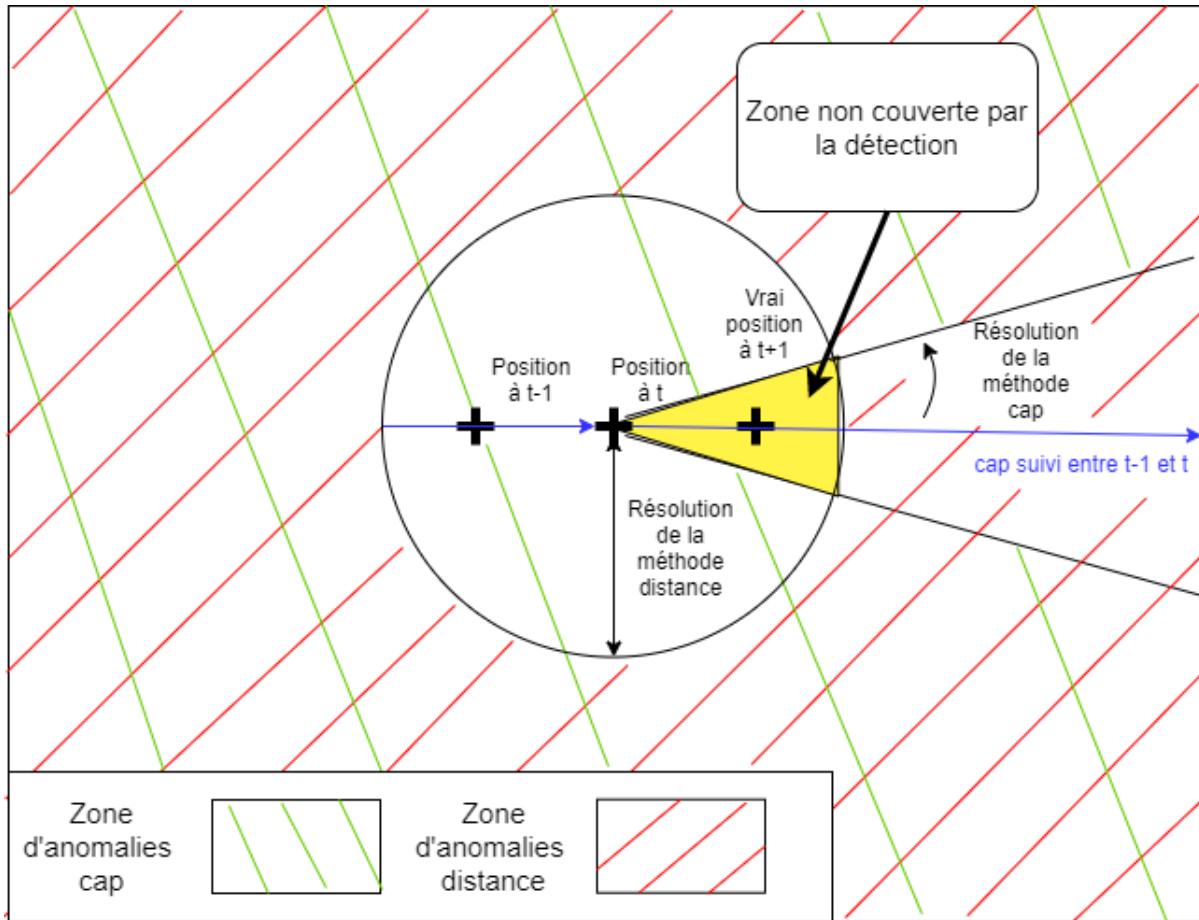


Figure 3.5 Zone de détection en combinant l'étude de la distance et variation de cap

Tests pour l'étude de la distance et résultats obtenus

Pour vérifier nos hypothèses, nous testons ce scénario en conditions virtuelles en modélisant une attaque interne grâce à l'association de "Bridge Command" et "Open CPN". Cela à pour but de vérifier si une attaque a bien un effet sur les données de cap, vitesse et distance, pour vérifier si l'étude de ces données est pertinente. L'analyse ne se fait pas pour l'instant en temps réel mais après coup, à partir d'une capture. Cette manipulation à pour but de déterminer si le protocole pourrait permettre une détection, en cherchant si une attaque se traduit par un comportement particulier détectable dans les données, que l'on pourrait alors rechercher pour détecter une attaque. Pour analyser les données, nous utilisons un outil de

data mining, "Orange Data Mining"[26], afin de représenter graphiquement les données et trouver des indices accusant le décalage.

Dans un premier temps, les résultats présentés sur la figure 3.6, représentant des données de distance en fonction de la vitesse fond, confirment ce qui était prévu : la distance consécutive et la vitesse fond ont bien une tendance proportionnelle. De plus, on observe une répartition sous forme de paliers de distance, provenant des arrondis lors du calcul de la distance sur "Orange Data Mining" (deux points consécutifs sont très proches vu la fréquence des trames).

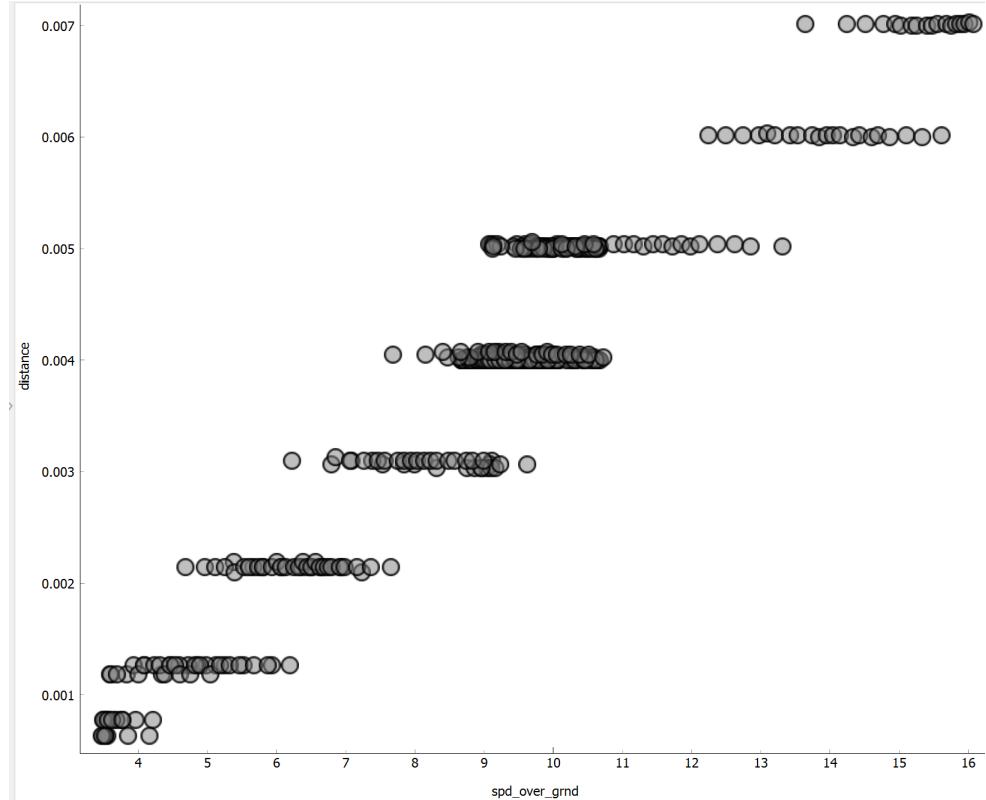


Figure 3.6 Graphique de la distance entre deux points successifs en fonction de la vitesse fond sans leurrage.

Ensuite, dans le cas où l'on commence à appliquer des décalages de différentes valeurs, on observe comme le montre la figure 3.7 des points isolés, qui montrent une incohérence entre distance et vitesse à ce moment t , qui correspondent au saut de début de décalage. A l'œil nu, ces points accusant un décalage du à un leurrage, sont clairement visibles. Le but est maintenant de créer un programme pour détecter la présence de ce genre de points automatiquement et en temps réel, surtout lorsque le décalage devient faible.

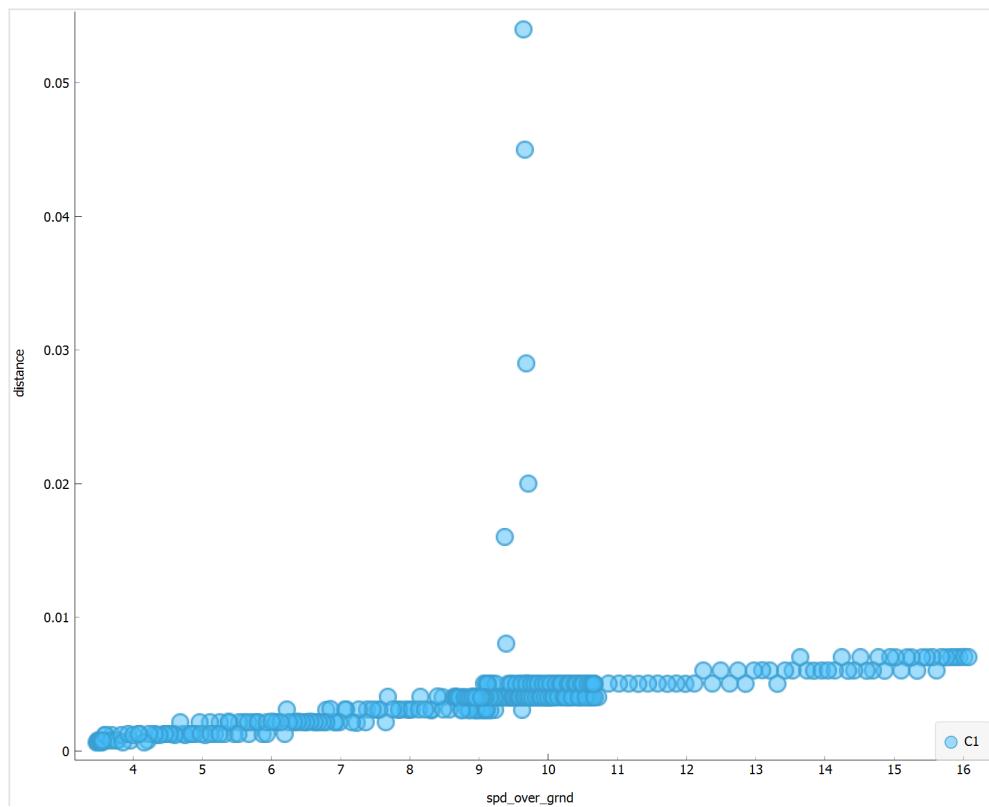


Figure 3.7 Graphique des de la distance entre deux points successifs en fonction de la vitesse fond avec différentes valeurs de décalage à des temps différents

3.2 Méthodes d'apprentissage et implémentation

L'utilisation de "*Orange data mining*" à permis de montrer qu'il est possible d'étudier les anomalies dans les données de cap et distance pour détecter des leurrages de position. Pour

cela, il faudrait, en temps réel, vérifier que les données de cap et de distance du bâtiment correspondent à des données considérées normales ou non. Nous allons utiliser des méthodes de détection, d'abord statistiques puis basées sur de l'apprentissage automatique, notamment de *novelty detection*, les valeurs de distance et cap étant alors utilisées comme *features*.

3.2.1 Méthode statistique

Nous avons d'abord choisi d'utiliser une méthode, dite statistique. Cette méthode n'est pas une méthode d'apprentissage automatique, mais se base sur les données précédentes, considérées comme provenant d'une situation sans leurrage, pour construire un modèle. On confronte ensuite au modèle des données produites en temps réel pour déterminer si celles-ci correspondent à une situation normale ou non.

Une première approche consiste à étudier, dans un premier jeu de données d'entraînement, la variable variation de cap entre deux points, notée dC , pour en extraire une moyenne μ et un écart-type σ . On calcule le cap du porteur en un point grâce au point précédent via les formules d'estime dynamique (apprises en cours à l'École Navale) en faisant l'hypothèse que ce cap est constant sur l'intervalle de temps entre deux points. Pour obtenir un modèle complet, on réalise plusieurs captures selon des paramètres différents de vitesse, d'agitation de la mer, d'angle de barre et de cap. On récupère pour toutes ces captures les différentes moyennes et écarts-type de la variable dC . La figure 3.8 présente une représentation graphique des résultats. On distingue deux points aberrants, ajoutés lors de la simulation après coup, sur la partie droite de la courbe que l'on classifie comme des anomalies

La phase de test consiste alors à évaluer pour une trame donnée la variable $Z = \frac{dC - \mu}{\sigma}$, avec dC la variation de cap observée. En faisant l'hypothèse que Z suit une loi normale tant qu'il n'y a pas de brouillage, on estime avec une probabilité de 99 % que Z se situe dans l'intervalle [-3 , +3]. On considère alors que la trame est sujette à un leurrage dès que $|Z| > 3$.

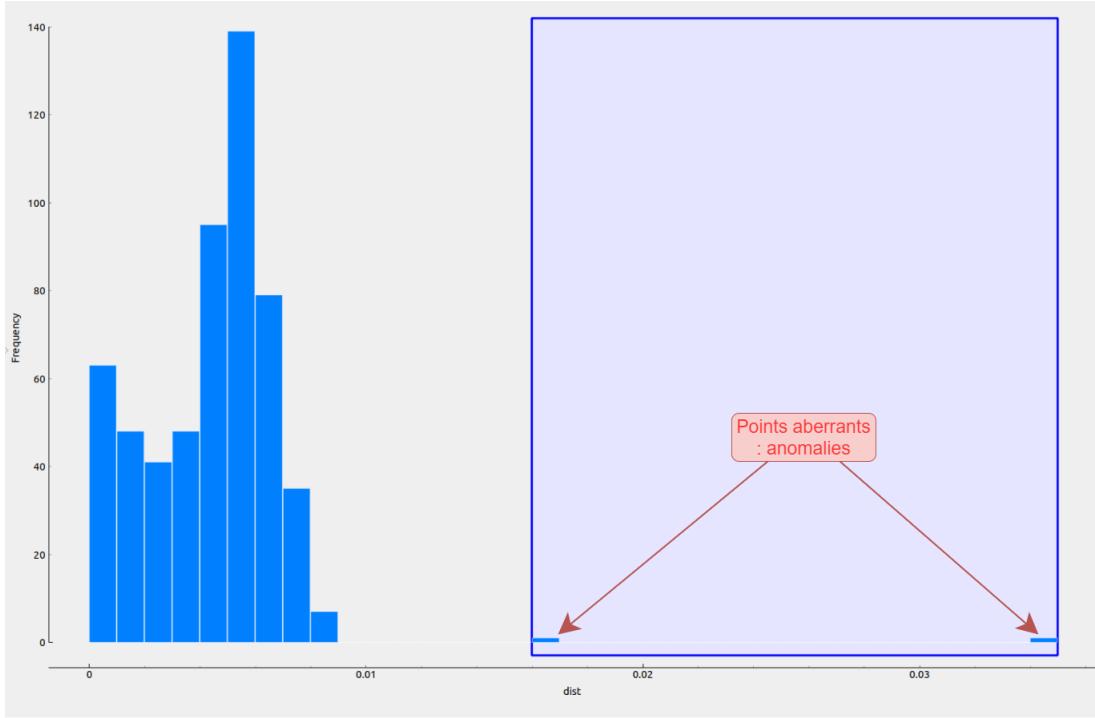


Figure 3.8 Répartition statistique de la distance entre deux points successifs normalisée

On réalise le même processus avec l'évaluation de la répartition statistique de la variation de distance entre deux points successifs, mais aussi les variations de latitude et de longitude.

Finalement, on réalise deux algorithmes de détection, l'un prenant comme *feature* la distance et le cap ensemble, et l'autre prenant simultanément les écarts de latitude et de longitude. Les résultats de ces algorithmes sont donnés dans la section "Résultats" ci-dessous.

3.2.2 Apprentissage automatique avec "*sklearn*"

La bibliothèque Python "*sklearn*" permet de réaliser des classifications grâce à l'apprentissage automatique en utilisant des méthodes différentes. L'objectif est alors d'identifier la méthode, ainsi que ses paramètres les plus adaptés, qui présentent les meilleurs résultats après application aux données présentées. Dans notre cas, nous cherchons une méthode qui puisse être entraînée à l'avance, puis être capable de détecter si un point présenté ensuite est con-

sidéré comme normal ou comme une anomalie. Cette situation est appelée en apprentissage automatique la *novelty detection*. Il est pertinent dans notre cas, puisque notre objectif est de détecter en temps réel des anomalies sur les trames [27][15]. Il convient de s'intéresser seulement aux algorithmes pouvant réaliser cette tâche. Pour utiliser "*sklearn*" dans le cas de la *novelty detection*, il convient de définir un ou des évaluateurs.

Ces évaluateurs sont des fonctions qui prennent en paramètre un jeu de données, dit d'apprentissage. Une première phase consiste donc à produire les données qui serviront de base d'entraînement pour les évaluateurs. Nous créons des données d'entraînement comme spécifié dans le paragraphe 2.2.1.

Cependant, les données d'entraînement ne sont pas utilisées telles quelles : une phase de pré-traitement (ou *preprocessing*) est nécessaire. D'abord, les données aberrantes sont supprimées. Aucune attaque n'a pour le moment été réalisée, ces point proviennent peut-être de problèmes survenus lors de la capture ou d'un bogue de "*Bridge Command*". Dans notre cas, par exemple, des distances négatives, ou des points vraiment isolés des autres (repérés grâce à une représentation graphique), sont supprimés. Sans cette étape de suppression, les évaluateurs construirait leur modèle, sensé ne représenter que des situations normales, à partir de points contenant déjà des anomalies, ce qui fausserait la classification.

Ensuite, les algorithmes de "*sklearn*" nécessitent que les données de départ soient normalisées pour appliquer un modèle. Nous utilisons donc la fonction de normalisation automatique de "*sklearn*" sur les données avant d'utiliser les évaluateurs. De plus, nous n'oubliions pas de normaliser les données ajoutées en temps réel de la même façon que les données d'entraînement. De même, il est important de prendre en compte le fait que les évaluateurs ont des méthodes de construction et paramètres propres à chaque algorithme, dont dépend le modèle construit et donc l'efficacité. Pour commencer, un affichage visuel des modèles permet d'avoir une bonne idée de l'efficacité des évaluateurs. On choisit d'abord d'étudier, comme présenté dans "*Orange Data Mining*", la distance parcourue en fonction de la vitesse. Les données ont été pré-traitées. Ces points présentent des situations normales, l'algorithme

devra détecter une anomalie lorsque qu'un nouveau point est situé en-dehors de la frontière construite. Un point avec décalage (de couleur violette) est ajouté *a posteriori* sur les figures, après construction du modèle, pour vérifier que ce dernier n'est pas identifié comme normal. On cherche à voir si l'algorithme apprend une frontière qui englobe bien la majorité de ces points normaux, frontière qui marque la séparation entre point normaux et anomalies. Par exemple, même après ajustement des paramètres, la méthode *Local Outlier Factor* (LOF) présente une zone des points normaux qui paraît tout de suite problématique (voir figure 3.9) avec les paramètres employés. Sur cette figure le point blanc largement au dessus des autres est un point aberrant des données d'entraînement, les modèles ne le prennent pas en compte. Le point violet est une anomalie ajoutée après entraînement.

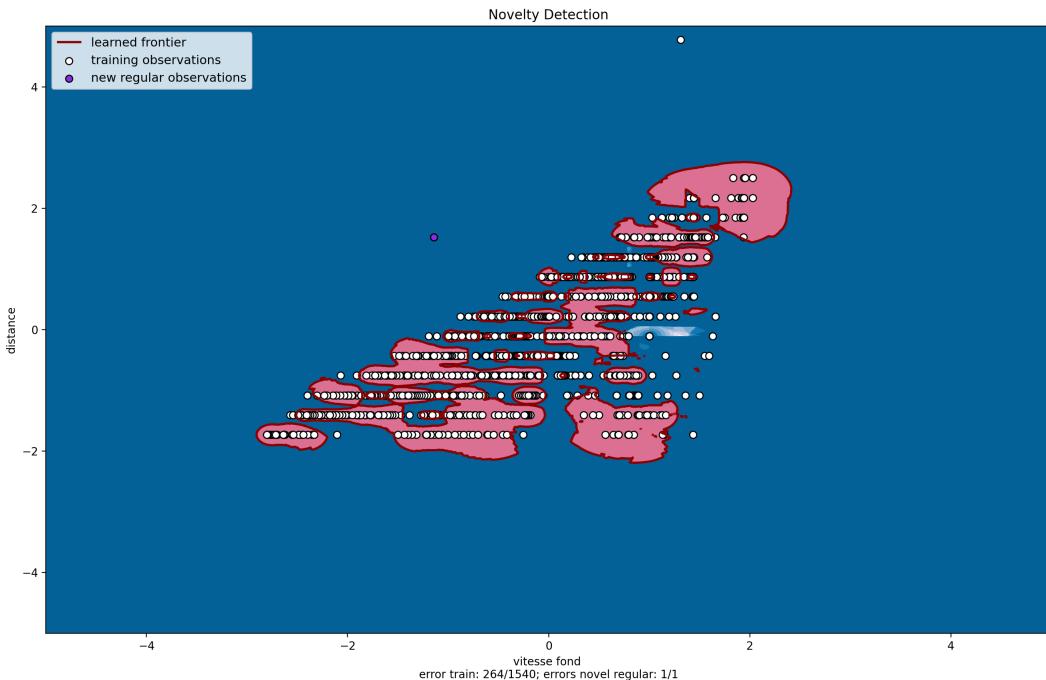


Figure 3.9 Essai de classification de données normales avec l'algorithme LOF.

Par opposition, un algorithme de type *Support Vector Machine* (SVM), semble plus prometteur, les données étant visuellement mieux englobées (*cf* figure 3.10).

Cependant, ces méthodes visuelles sont approximatives, nous devons pouvoir comparer

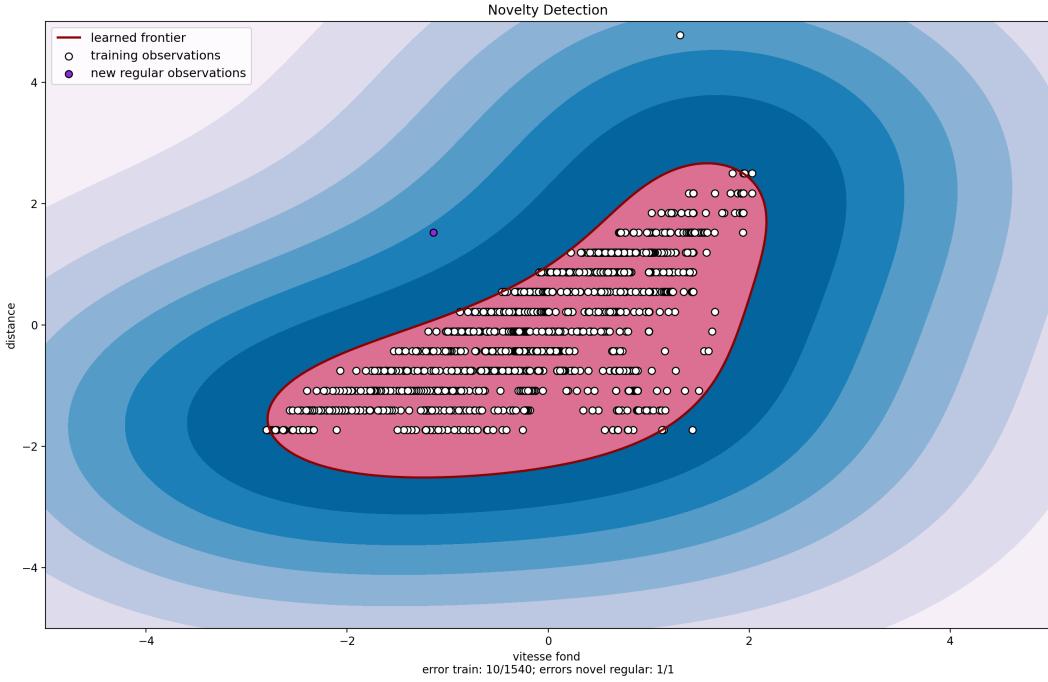


Figure 3.10 Premier essai de classification de données normales avec l'algorithme SVM

les méthodes "sklearn" entre elles, avec différents paramètres, ainsi qu'avec la méthode statistique de façon plus rigoureuse, afin de déterminer laquelle est la plus efficace pour détecter des leurrages.

3.2.3 Score et implémentation

Pour confirmer ces résultats et le choix de la méthode, nous choisissons de simuler à la fois des attaques internes et externes : d'utiliser "*Bridge Command*" et notre outil *man in the middle* pour simuler une attaque interne pour établir la plupart des résultats et de les confirmer en testant les programmes sur l'eau ensuite avec une attaque externe. Afin d'évaluer le score de chacune des méthodes avec l'attaque interne, on simule une situation complète de navigation avec "*Bridge Command*" (lignes droites, girations et changements d'allure). Le script Python modifie une proportion réglable de trames en temps-réel et choisies aléatoirement avec un

		Statistique ϕ et G	Statistique cap et distance	LOF	SVM
100 yards	Score de classification	89%	100%	98%	100%
	Faux-positifs	1%	0%	0%	0%
10 yards	Score de classification	86%	99%	97,4%	99,2%
	Faux-positifs	2%	1%	0%	0%
1 yards	Score de classification	74%	89%	87,8%	91,7%
	Faux-positifs	3%	2%	0%	0%

Table 3.1 Tableau présentant les scores des évaluateurs en fonction de la valeur des décalages appliqués.

décalage, lui aussi réglable. Toutes les trames sont ensuite analysées et classifiées comme trame normale ou comme anomalie par plusieurs versions de chacun des algorithmes de détection et en fonction des différents paramètres. Afin de déterminer le score de chacun, on compte le nombre de bonnes prédictions, c'est à dire les cas de vrai-positif (leurrage et détection) et faux-négatif (pas de leurrage et pas de détection), et on divise par le nombre total de trames, ce qui donne la proportion de trames bien classifiées. La proportion de faux-positifs (lorsque l'algorithme détecte une anomalie alors que la trame n'a pas été modifiée), est aussi vérifiée. Reste alors les cas de vrai-négatif, c'est à dire les cas où un leurrage n'est pas détecté (dans notre cas, cela correspond à l'inverse de l'union du score et de la proportion de faux positif). On teste ce protocole de score avec différentes valeurs de décalage. On fixe arbitrairement la proportion de trames modifiées à 10%.

Après plusieurs tests et conditions initiales, on remarque que la méthode *Support Vector Machine* est la méthode qui obtient les meilleurs scores. Le tableau 3.1 résume les résultats des tests, avec des décalages différents, avec les meilleures versions de LOF et SVM (avec les paramètres ayant obtenu les meilleurs scores).

Nous décidons de vérifier les résultats de la méthode SVM en parallèle lors d'expériences

avec l'embarcation à deux reprises, pendant la phase de développement et pendant la phase finale. Lors de la première expérience seulement, le programme étant alors en développement, seules les données de distances sont testées. Le protocole de test est le même que celui décrit en partie II. Comme nous n'avons pas pu accéder pendant longtemps au matériel, les évaluateurs sont entraînés seulement avec des données de "*Bridge Command*". Dans "*Bridge Command*", nous avons choisi le navire qui se rapproche le plus de l'embarcation que nous utilisons dans la réalité, et avons effectué aussi une capture avec diverses situations de navigation (différentes vitesses et girations). Afin de recréer une situation de navigation qui s'approche de la réalité avec notre embarcation, nous planifions des circuits à effectuer avec l'embarcation dans les deux sens, contenant notamment des lignes droites et des virages des deux côtés (voir figure 3.11). Des leurrages sont effectués par le PLUTO, avec différentes valeurs de décalage (voir figure 3.12). Un changement d'allure à des moments aléatoires sont aussi effectués par le pilote.

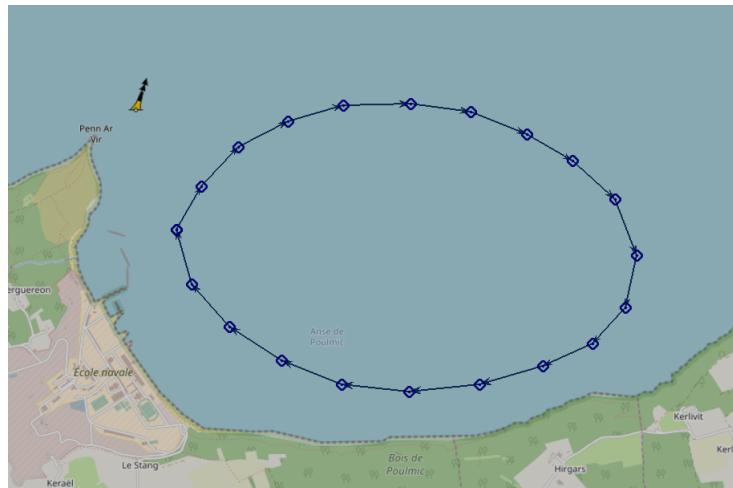


Figure 3.11 Exemple de circuit de test des évaluateurs avec l'embarcation.

Si les deux expériences montrent que le programme SVM obtient des résultats satisfaisants (score de 90%, puis 95%, avec un décalage de 10 yards), elles auront aussi permis de mettre en lumière un certain nombre de limites qu'il est important de prendre en compte.

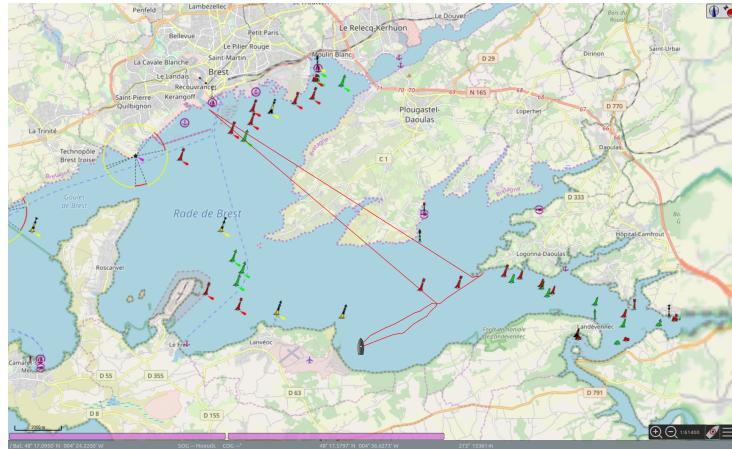


Figure 3.12 Exemple de leurrage effectué par le PLUTO

3.2.4 Affichage de la classification

Maintenant que la meilleure méthode de détection à été trouvée, l'idée est d'exploiter la classification du programme et d'informer le navigateur de la détection d'un leurrage. Pour cela, nous avons pensé qu'utiliser directement l'écran de l'ECDIS était la solution d'alerte la plus adaptée, cet outil étant par conception un outil de fusion et d'alerte. De cette manière, l'officier de quart en passerelle est alerté rapidement et peut prendre des mesures pour continuer à naviguer sainement. Pour transmettre l'information issue de la classification du programme, de nouveaux *sentence IDs* NMEA ont été créés, pour que la transmission de cette information suive le même standard de transmission que les autres informations à bord. Ces trames contiennent une *sentence ID* propre (\$CY pour cyber) un champ de temps (pour la correspondance avec le temps GPS), un champ indiquant la classification des trames GPS (0 pour normale, 1 pour anomalie) et une somme de contrôle (voir figure 3.13). Ces trames sont ensuite interprétées par l'ECDIS pour afficher un repère visuel sur l'écran, changeant de couleur si une anomalie est détectée (voir figure 3.14, le vert étant affiché en cas de situation normale et rouge lorsque des anomalies sont détectées).

En l'état, cet afficheur visuel n'utilise que les programmes que nous avons créés. Il serait possible de le compléter en lui permettant d'afficher les soupçons de brouillage en

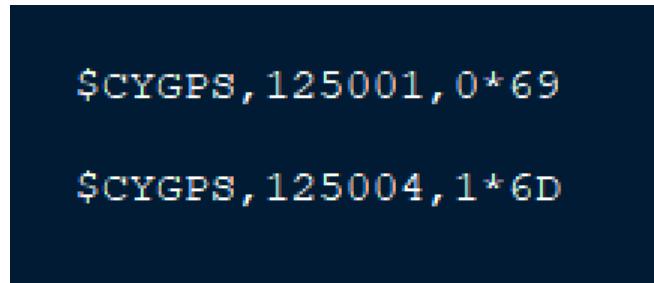


Figure 3.13 Exemple de trames \$CY.

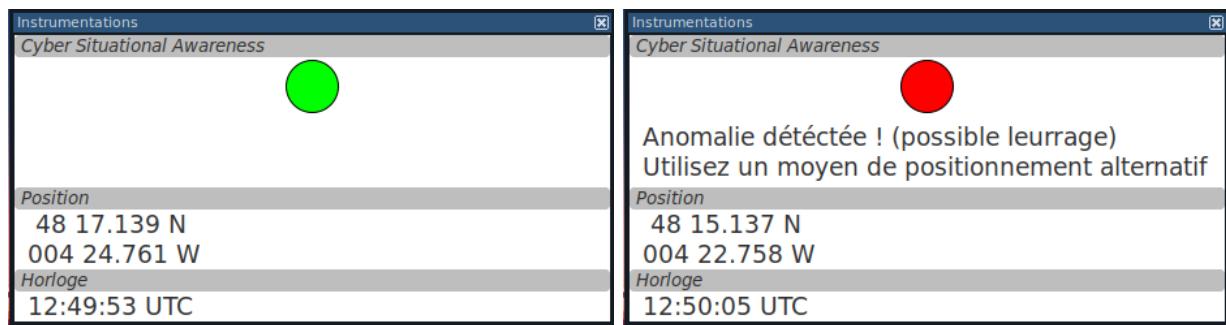


Figure 3.14 Images du rendu visuel de l'indicateur sur l'ECDIS.

complément des alertes de leurrage. Comme évoqué dans la partie II, les systèmes navals sont déjà capables d'afficher la perte d'information GPS survenant lors d'un brouillage, il faudrait alors lier les deux pour disposer d'un outil complet. Cependant, le navigateur doit garder un œil critique sur les informations qu'il voit à l'écran, la classification n'est pas absolue et des erreurs demeurent possibles.

3.3 Limites et critiques de la méthodologie de détection et des résultats

Bien que les résultats semblent être satisfaisants, il est important d'avoir conscience des limites de notre programme pour l'utiliser à bon escient. Il est possible de séparer les limites que nous avons identifiées en deux catégories : celles issues de notre méthode de détection des leurrages et du contexte maritime en lui-même, et celles liées aux algorithmes.

3.3.1 Limites de la méthode

En plus de confirmer les résultats en situation réelle, l'expérience à bord des embarcations à aussi permis de montrer des limites et spécificités propres à l'application au domaine maritime du programme.

Résolution

D'abord, il est important de souligner que la détection des décalages a une résolution fixée. En effet, la détection des décalages devient de plus en plus difficile lorsque ceux-ci deviennent petits, comme le confirment les scores des programmes. Pourtant, une suite de décalages, même petits, peut au final entraîner un écart important par rapport à la position réelle, pouvant même aller jusqu'à une divergence (voir la figure 3.2). Ce problème, avec notre méthode, semble complexe à résoudre : la résolution peut augmenter, mais ne sera jamais parfaite, d'autant plus que les appareils de navigation ont eux aussi une imprécision (de l'ordre de 0,7 mètres 95% du temps pour la position GPS [7]), et que des éléments de navigation réels peuvent aussi influer sur les résultats (courant et vent entre deux positions consécutives par exemple).

Leurrage par fixation

Par la suite, l'expérience nous aura permis d'identifier un cas de leurrage qui ne pourrait pas être détecté par notre méthode. En effet, notre méthode exploite la détection de décalages entre des positions consécutives. Or, un leurrage qui fixe le porteur sur une position donnée n'impliquerait qu'un décalage au début et à la fin du leurrage, voire même pas de décalage du tout si la position choisie est celle du bâtiment à l'instant t (voir figure 3.15). La distance et la variation de cap entre deux points consécutifs sont alors nulles pendant le leurrage. En l'état, aussi bien en simulation que sur l'embarcation, notre programme est incapable de détecter ce type de leurrage (il ne détecte que les décalages éventuels au début ou à la fin).

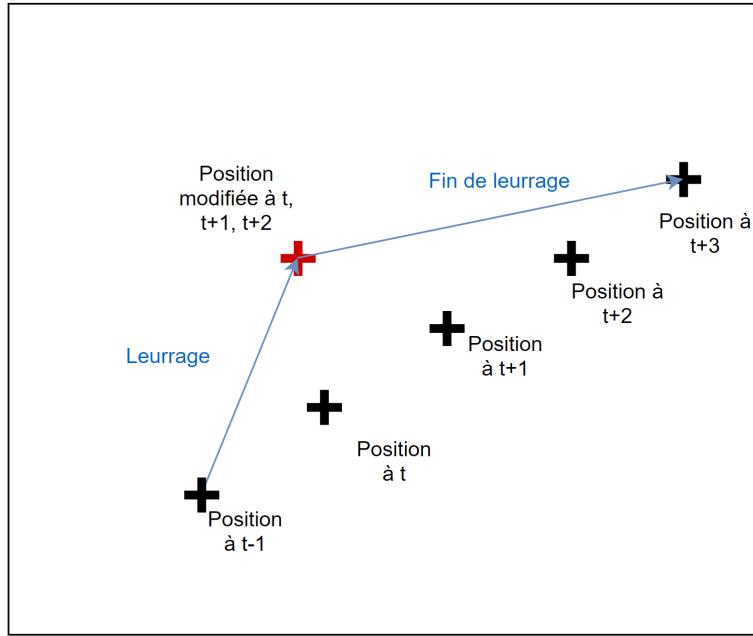


Figure 3.15 Exemple de cas de leurrage où la position est fixée.

On pourrait se dire qu'une vitesse trop élevée pour un décalage nul pourrait être détectée par notre programme comme étant une anomalie, mais ce n'est pas le cas : la figure 3.10 montre bien des points normaux qui ont une distance faible même à grande vitesse. Rappelons tout de même que ces données d'entraînement sont issues d'un simulateur : nous pensons que les changements d'allure dans la réalité sont moins brutaux, et donc qu'il est peu probable d'avoir des cas non leurrés de distance faible avec une grande vitesse. Obtenir plus de données issues de cinématiques de navires en opérations permettraient de détecter plus de cas de leurrage et ce d'autant plus que les cinématiques dépendent également fortement du type de navire et de ses missions.

3.3.2 Biais de l'apprentissage automatique

L'apprentissage automatique nous aura permis d'obtenir de bon résultats de détection et cela en temps-réel. Cependant, ces techniques et algorithmes ne sont pas exempts de biais. Il est indispensable de conserver un œil critique sur les résultats obtenus et de vérifier, à

chaque étape d’implémentation logicielle et de tests, qu’un biais n’est pas présent. Dans notre contexte, nous avons identifié les biais suivants comme pouvant avoir un impact sur nos résultats [28].

Biais d’échantillon

Ce premier biais s’applique aux données d’entraînement à proprement parler. Pour que le modèle construit par le programme fonctionne, le bon choix des données d’entraînement est primordial. En effet, elle doivent correspondre à l’ensemble des situations normales pour éviter les faux-positifs : si l’ensemble des situations n’est pas décrit, l’algorithme pourrait considérer des situations normales comme étant des anomalies. Cela implique, dans notre contexte, d’entraîner notre programme avec des données correspondant à une large majorité de situation de navigation pour réduire ce biais (différentes cinématiques, météo...), ce que nous n’avons pas pu faire faute de temps et de moyens. Dans le cas d’un contexte opérationnel, un biais d’échantillons peut aussi être créé volontairement par un adversaire pour saboter les résultats, il s’agit alors d’une attaque généralement empoisonnement des données (*adversarial AI data poisoning*). En résumé, l’objectif est d’avoir des données d’entraînement justes et qui correspondent à tous les cas de la réalité.

Biais d’exclusion

Un autre biais peut apparaître lors du pré-traitement de ces mêmes données d’entraînement, lors de la phase d’élimination des points aberrants. En effet, la suppression de ces points aberrants est, dans notre étude, arbitraire. Or, il se peut que certains de ces points représentent bien une situation normale, peu courante, mais possible. Si le bâtiment se retrouve dans la même situation, le programme risque de détecter une anomalie, alors que la situation est normale, entraînant un faux-positif. A l’inverse, il se peut que certains points aberrants soient conservés, et donc que le modèle construit soit mauvais. Une étude approfondie des données d’entraînement est nécessaire pour obtenir un modèle performant.

Biais de mesure

Enfin, pour que la classification se fasse correctement, les données obtenues en temps-réel doivent provenir de mesures réalisées dans des conditions similaires aux mesures des données d'entraînement. En effet, si ce n'est pas le cas, les données à classifier sont confrontées à un modèle qui peut être différent et qui peut ne pas s'appliquer à ces données. Dans notre cas, nous avons utilisé pour tester les trames de l'embarcation un modèle créé à partir de données d'une source différente : "*Bridge Command*". Cela a inévitablement créé un biais de mesure.

Conclusion

Réalisation finale et bilan

Pour conclure, ce projet nous aura permis de comprendre le standard NMEA, son importance dans la conduite du navire et ses vulnérabilités, puis de mettre en place des méthodes de détection de leurrage GPS en utilisant des techniques de *machine learning*. Ce projet s'inscrit dans un cadre plus global puisque notre projet a été intégré avec les autres axes de recherche de la Chaire de cyberdéfense des systèmes navals. Nous avons notamment pu entrevoir une partie plus "offensive" avec la réalisation de leurrage GPS lors des mises à l'épreuve de nos modèles.

A l'issue de ce projet, il est possible de proposer plusieurs perspectives de recherche. Tout d'abord, nous avons fait le choix de restreindre le champ de notre étude aux seules valeurs de position (longitude et latitude) au sein d'une trame NMEA 0183, sans prendre en compte le temps qui demeure cependant une donnée fondamentale pour les systèmes. Des travaux complémentaires pourraient ainsi être menés pour permettre la détection d'anomalie temporelle.

Ensuite, nous avons concentré notre analyse sur le contenu des trames réseau. Il aurait également été possible de prendre en compte certaines métadonnées des trames. En effet, l'analyse de la fréquence d'émission des trames et des ports d'émission et de réception auraient peut-être permis de détecter d'autres attaques, notamment internes.

Par ailleurs, la corrélation des informations du GPS avec d'autres capteurs comme le

RADAR, l'AIS et autres systèmes de positionnement par satellites présenterait un intérêt réel pour affiner les détections.

Enfin, les trames GPS ne constituent qu'une fraction du trafic total des trames NMEA 0183, et de nombreux autres types de trames peuvent aussi être la cible d'attaque. Nous avons tenté d'élargir notre étude à d'autres types de *sentences ID* différents (notamment \$RPM, pour la vitesse de rotation des moteurs), mais nous avons été rapidement bloqués par les approximations faites lors de la création de ces trames par "*Bridge Command*" (la montée en allure des machines est instantanée, ce qui est différent de la réalité. On ne peut donc pas détecter de saut comme nous l'avons fait pour la distance et le cap par exemple). Une version de "*Bridge Command*" plus complète et proche de la réalité est actuellement en développement à la Chaire de cyberdéfense de l'École Navale.

Gestion de projet et apports

Finalement, le projet aura été mené à bien, malgré les conditions de travail dégradées en raison de la pandémie de COVID-19, et les multiples coupures dues aux permissions et à la corvette Gants Blancs. Grâce notamment à l'outil RENATER, nous avons maintenu des séances de travail optimales en viso-conférence, régulières et cohérentes avec le planning, en plus de visites en présentiel dans les locaux de la chaire, deux à trois fois par semaine quand cela était possible. Deux sorties sur le plan d'eau ont aussi pu être réalisées malgré la crise. (voir figure 3.17).

Nous avons pu bénéficier de l'encadrement de la Chaire de cyberdéfense des systèmes navals, notamment en travaillant en étroite collaboration avec le doctorant Clet Boudehenn, qui a réalisé toute la partie offensive consacrée aux brouillages et leurrages GPS et sans qui nous n'aurions pas pu mettre à l'épreuve notre modèle.

Ce PFE nous aura permis d'approfondir la connaissance du domaine de cybersécurité du monde maritime que nous avions entrevu lors de notre projet de voie d'approfondissement le



Figure 3.16 Le doctorant Clet Boudehenn manipulant le SDR-ADALM-PLUTO lors d'une sortie en mer

semestre dernier. De plus, nous avons pu nous initier à l'utilisation concrète de techniques de l'apprentissage automatique.

Nous avons notamment eu la chance de pouvoir confronter notre modèle théorique avec la réalité du terrain en réalisant deux sorties en embarcation pneumatique.

Enfin, de par son application concrète, ce projet nous aura permis de faire le lien entre la formation scientifique délivrée à l'École Navale et notre futur travail d'Officier Chef de Quart, responsabilité qui sera la nôtre dès l'année prochaine sur les bâtiments de la Marine Nationale.

L'ensemble des productions et codes réalisés pour ce projet sont disponibles en ligne, sur le site "*Github*", PFE NMEA [29]

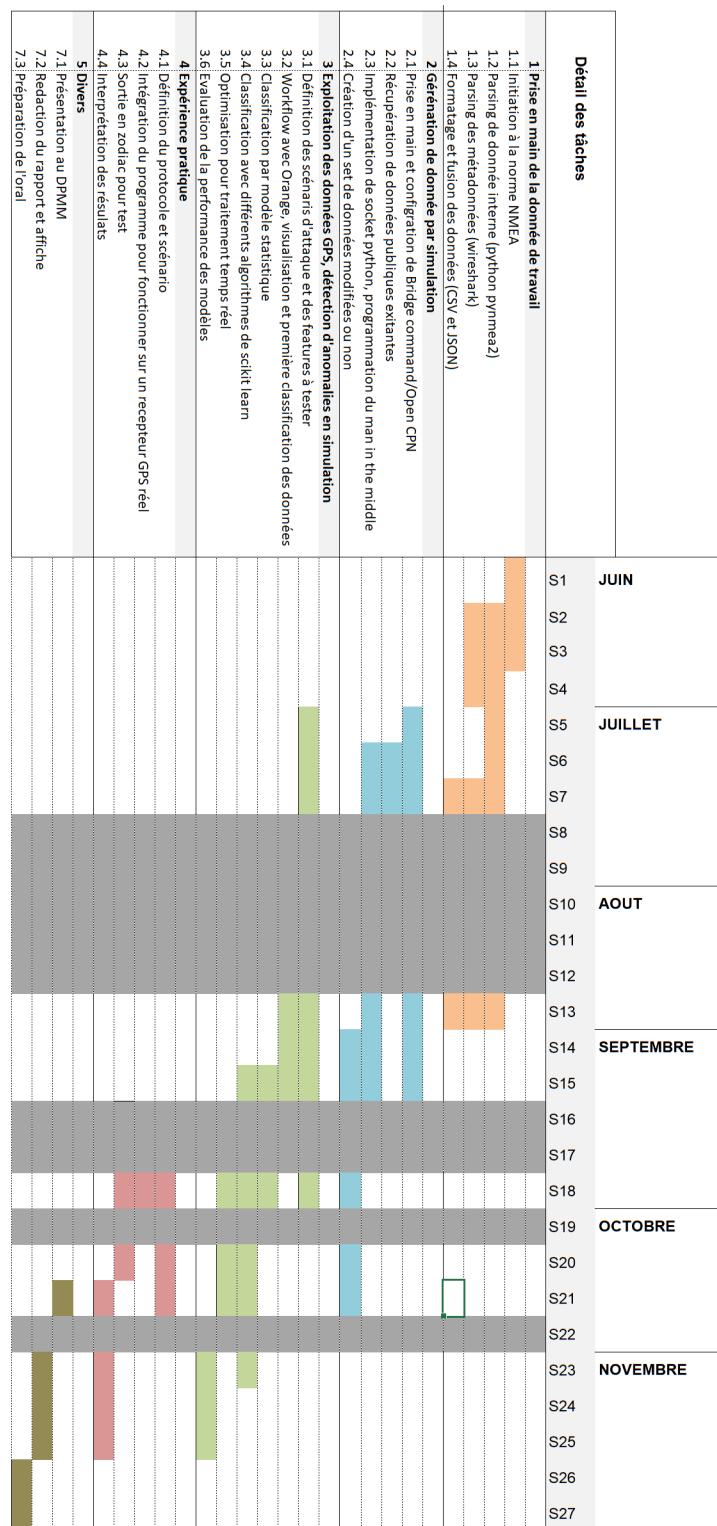


Figure 3.17 Diagramme de Gantt du Projet de Fin d'Études

Bibliographie

- [1] Site Internet de la *National Maritime Electronics Association*
<https://www.nmea.org/>
Consulté le 08/06/2020
- [2] Document précisant la configuration d'un réseau NMEA 0183
NMEA 0183 Standard For Interfacing Marine Electronic Devices Version 3.01 January 1, 2002
<http://www.plaisance-pratique.com/IMG/pdf/NMEA0183-2.pdf>
Consulté le 13/06/2020
- [3] Calculateur de *checksum*
<https://nmeachecksum.eqth.net>
Consulté le 24/06/2020
- [4] Exemple de cas d'utilisation de matériel piégé
<https://threatpost.com/on-board-mystery-boxes-threaten-global-shipping-vessels/149211/>
Consulté le 02/07/2020
- [5] Quelques exemples de risques induits par l'usage du GPS
<https://cybermaretique.fr/les-risques-cyber-lies-aux-moyens-de-positionnement/>
Consulté le 20/06/2020
- [6] Ephémérides des satellites
<https://cddis.nasa.gov/archive/gnss/data/daily/2020/315/20n/brdc3150.20n.Z>
Consulté le 24/09/2020
- [7] Site informatif officiel du gouvernement américain pour le GPS
<https://www.gps.gov/>
<https://www.gps.gov/multimedia/poster/poster-web.pdf>
<https://www.gps.gov/applications/marine/>
<https://www.gps.gov/systems/gps/performance/accuracy/>

<https://www.gps.gov/applications/timing/>
Consulté du 10/06/2020 au 02/11/2020
- [8] Théorie du brouillage et leurrage GPS
<https://www.maritimeglobalsecurity.org/media/1043/2019-jamming-spoofing-of-gnss.pdf>
Consulté le 04/07/2020

BIBLIOGRAPHIE

- [9] Rapport sur les incidents GNSS imputables à la Fédération de Russie
<https://c4ads.org/s/Above-Us-Only-Stars.pdf> Consulté le 13/07/2020
- [10] Exemple d'attaques cyber du monde maritime
Détection, analyse contextuelle et visualisation de cyber-attaques en temps réel : élaboration de la cyber situational awareness du monde maritime
Jacq Olivier, 2020
- [11] Exemple de rapport d'attaque NMEA
<https://www.pentestpartners.com/security-blog/crashing-ships-by-hacking-nmea-sentences/>
Consulté le 01/07/2020
- [12] Exemple de cas de leurrage : le Stena Impero
<https://www.gpsworld.com/iran-jams-gps-on-ships-in-strait-of-hormuz/>
Consulté le 03/07/2020
- [13] Schéma de principe du *machine learning*
Clet Boudehenn - *Démystification du machine learning _centre_support_cyberdéfense.pdf*
Consulté le 28/08/2020
- [14] Exemples d'utilisation d'apprentissage automatique
Alpaydin Ethem,
Introduction to machine learning
2020, MIT press., page 4
- [15] Etude sur l'utilisation de l'algorithme SVM pour la détection d'anomalie
Tolga Ergen and Suleyman S. Kozat
A novel distributed anomaly detection algorithm based on support vector machines , 2020, Digital Signal Processing, Volume 99
- [16] Un exemple de diagramme de sélection d'évaluateur, publié par la librairie Scikit Learn
https://scikit-learn.org/stable/_static/ml_map.png
Consulté le 18/09/2020
- [17] Principe de l'algorithme LOF
https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html
Consulté le 13/10/2020
- [18] Site de Bridge Command
<https://www.bridgecommand.co.uk/>
Consulté le 12/07/2020
- [19] Site de Open CPN
<https://opencpn.org/index.html>
Consulté le 13/07/2020
- [20] Ressource parser NMEA pour Python
<https://github.com/Knio/pynmea2>
Utilisé à partir 06/06/2020

BIBLIOGRAPHIE

- [21] Exemple de jeu de données NMEA disponible en ligne.
[https://catalog.data.gov/dataset/oceanographic-profile-chlorophyll
-a-and-zooplankton-biomass-measurements-collected-using-bottle](https://catalog.data.gov/dataset/oceanographic-profile-chlorophyll-a-and-zooplankton-biomass-measurements-collected-using-bottle)
Consulté le 03/06/2020
- [22] Stratégie nationale de sûreté des espaces maritimes française
[https://www.gouvernement.fr/sites/default/files/
contenu/piece-jointe/2019/12/snsem_2019_finale.pdf](https://www.gouvernement.fr/sites/default/files/contenu/piece-jointe/2019/12/snsem_2019_finale.pdf)
Consulté le 03/09/2020
- [23] Caractéristiques techniques ADALM-PLUTO
[https://www.analog.com/media/en/news-marketing-collateral/
/product-highlight/ADALM-PLUTO-Product-Highlight.pdf](https://www.analog.com/media/en/news-marketing-collateral/product-highlight/ADALM-PLUTO-Product-Highlight.pdf)
Analog Devices, 2017, Consulté le 06/10/2020
- [24] Exemple spécification d'un GPS haute fréquence
[http://nke-marine-electronics.fr/wp-content/uploads/2017/01/
high-frequency-GPS.pdf](http://nke-marine-electronics.fr/wp-content/uploads/2017/01/high-frequency-GPS.pdf)
Consulté le 03/06/2020
- [25] Spécification des fréquences du service GPS
<https://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf>, page 14
United States Coast Guards, 2019, Consulté le 03/06/2020
- [26] Site Internet du logiciel Orange *data mining*
<https://orange.biolab.si/>
Consulté le 17/09/2020
- [27] Description des méthodes de détection d'anomalies disponible sur sklearn.
https://scikit-learn.org/stable/modules/outlier_detection.html Consulté le 13/10/2020
- [28] Shepperd, Martin and Bowes, David and Hall, Tracy
Researcher bias: The use of machine learning in software defect prediction
IEEE Transactions on Software Engineering
pages 603 à 616, 2014 Consulté le 23/10/2020
- [29] Lien vers la plateforme "*Github*" du projet
https://github.com/CHEVALLIERCYBER/PFE_NMEA