Version française abrégée de la thèse :

ANALYSE DE RISQUE ET DETECTION D'INTRUSIONS POUR LES RESEAUX AVIONIQUES

Silvia GIL CASALS

ANALYSE DE RISQUE ET DETECTION D'INTRUSIONS POUR LES RESEAUX AVIONIQUES

RÉSUMÉ

L'aéronautique connaît de nos jours une confluence d'événements: la connectivité bord-sol et au sein même de l'avion ne cesse d'augmenter afin, entre autres, de faciliter le contrôle du trafic aérien et la maintenabilité des flottes d'avions, offrir de nouveaux services pour les passagers tout en réduisant les coûts. Les fonctions avioniques se voient donc reliées à ce qu'on appelle le Monde Ouvert, c'est-à-dire le réseau non critique de l'avion ainsi qu'aux services de contrôle aérien au sol. Ces récentes évolutions pourraient constituer une porte ouverte pour les cyber-attaques dont la complexité ne cesse de croître également. Cependant, même si les standards de sécurité aéronautique sont encore en cours d'écriture, les autorités de certification aéronautiques demandent déjà aux avionneurs d'identifier les risques et assurer que l'avion pourra opérer de façon sûre même en cas d'attaque.

Pour répondre à cette problématique industrielle, cette thèse propose une méthode simple d'analyse de risque semi-quantitative pour identifier les menaces, les biens à protéger, les vulnérabilités et classer les différents niveaux de risque selon leur impact sur la sûreté de vol et de la potentielle vraisemblance de l'attaque en utilisant une série de tables de critères d'évaluation ajustables. Ensuite, afin d'assurer que l'avion opère de façon sûre et en sécurité tout au long de son cycle de vie, notre deuxième contribution consiste en une fonction générique et autonome d'audit du réseau pour la détection d'intrusions basée sur des techniques de Machine Learning. Différentes options sont proposées afin de constituer les briques de cette fonction d'audit, notamment : deux façons de modéliser le trafic au travers d'attributs descriptifs de ses caractéristiques, deux techniques de *Machine Learning* pour la détection d'anomalies : l'une supervisée basée sur l'algorithme *One Class Support Vector Machine* et qui donc requiert une phase d'apprentissage, et l'autre, non supervisée basée sur le *clustering* de sous-espace. Puisque le problème récurrent pour les techniques de détection d'anomalies est la présence de fausses alertes, nous prônons l'utilisation du *Local Outlier Factor* (un indicateur de densité) afin d'établir un seuil pour distinguer les anomalies réelles des fausses alertes.

Ce mémoire rend compte du travail effectué dans le cadre d'une convention CIFRE (Convention Industrielle de Formation par la Recherche) entre THALES Avionics et le CNRS-LAAS de Toulouse.

Mots-clés: sécurité des réseaux avioniques, analyse de risque, processus, détection d'anomalies/d'intrusions, *Machine Learning*

TABLE DES MATIÈRES

CHAPITRE 5 : FONCTION DE DÉTECTION D'INTRUSION AUTONOME POUR LES RÉSEAUX AVIONIQUES

CHAPITRE 6 : CONCLUSION

CHAPITRE 1 : INTRODUCTION

1.1. MOTIVATION ET PROBLEMATIQUE

L'aéronautique fait face actuellement à une confluence d'événements. D'une part, la connectivité bord-sol et au sein même d'un avion ne cesse de croître, ce qui mène à la définition de nouvelles architectures, alors que les processus pour prendre en compte les aspects sécurité lors du développement sont encore pauvres voire inexistants. D'autre part, la quantité et la complexité des cyber-attaques ne cessent d'augmenter aussi et les démonstrations de vulnérabilités sont rendues publiques, mais pour l'instant, les standards de sécurité aéronautiques n'ont pas encore été publiés. Cependant, l'EASA[1] et la FAA[2], respectivement les autorités de certification européennes et américaines, ont émis des *Certification Review Items* (CRIs) et des *Special Conditions*[3] (SCs) aux avionneurs avec des aspects additionnels à être pris en compte lors de la certification concernant la protection contre les actes malintentionnés au niveau des systèmes et des réseaux. Répondre à ces requêtes est d'autant plus obligatoire que cela conditionne la délivrance du certificat de navigabilité de l'avion[4]. On y trouve généralement les aspects suivants :

1. Il faut assurer que les systèmes avioniques sont protégés de l'accès par des sources non-autorisées puisque leur corruption peut nuire à la sûreté du vol.
2. Les menaces portant sur l'avion (en incluant ceux causes pendant les opérations de maintenance ou par la connexion d'équipements non homologues) doivent être identifiées, évaluées et des stratégies pour réduire le risque doivent être mises en place pour protéger les systèmes de l'avion de tout impact adverse sur la sûreté du vol.
3. Il faut assurer que l'avion est maintenu dans des conditions de sûreté et de sécurité nécessaires tout au long de son cycle de vie.

Ces CRIs et SCs ont ont fourni les problématiques initiales qui ont inspire cette thèse sur la sécurité pour la sûreté des systèmes avioniques. Elle se focalise sur deux sujets principaux qui sont :

- Aspect processus. La définition d'une méthodologie d'analyse de risqué en accord avec les futurs standards de sécurité aéronautiques en cours de construction et compatible avec le processus de développement industriel en place pour répondre au point n°2.
- Aspect technique. Le design et la validation d'une fonction générique et autonome d'audit du réseau base sur des techniques de *Machine Learning* pour la détection d'intrusions et la caractérisation d'anomalies potentiellement causées par des cyber-menaces afin d'assurer que l'avion est maintenu dans des conditions de sécurité suffisantes tout au long de son cycle de (point n° 3).

[1] *European Aviation Safety Agency*

[2] *Federal Aviation Administration*

[3] Un exemple de *Special Condition* se trouve à l'adresse suivante : https://www.federalregister.gov/articles/2013/11/18/2013-27343/special-conditions-boeing-model-777-200--300-and--300er-series-airplanes-aircraft-electronic-system

[4] Certifie que l'avion est apte au vol et que le modèle de l'avion a été implémenté suivant un design approuvé au préalable en accord avec les standards et que toutes les mesures ont été prises pour assurer que l'avion est sûr. Ce certificat est obligatoire pour que l'avion soit autorisé à voler.

1.2. CHALLENGES ET CONTRIBUTIONS

1.2.1. EN TERMES DE PROCESSUS

Dans l'industrie des systèmes complexes telle que l'aéronautique, les processus sont cruciaux pour assurer l'interfaçage et la synchronisation adéquats entre les équipes travaillant sur les différents tronçons d'un avion et les activités transverses telles que la qualité, le processus de sûreté et la certification, et ce, tout au long des étapes de son cycle de développement (design, implémentation, vérification et validation). La sécurité est un tout nouveau domaine pour l'aéronautique, son but est de prévoir les éventuelles menaces sur les systèmes, ainsi que les potentiels cas de mauvaise utilisation afin d'éviter toute conséquence néfaste sur la sûreté de vol. Il ne faut pas confondre la « sécurité » avec la « sûreté », qui lui est un processus aéronautique, bien établi depuis plus de 50 ans, qui analyse les défaillances intrinsèques des systèmes. Il y a trois standards en cours de construction par des comités de l'EUROCAE (Europe) et du RTCA (USA) respectivement : DO-326A/ED-202A qui fournit les spécifications du processus de sécurité, DO-YY3/ED-203 qui donnera les méthodes et outils pour atteindre les objectifs de ce processus (notamment concernant les méthodologies d'analyse de risque) et DOYY4/ED-204 qui liste les instructions pour assurer le maintien en conditions de sécurité de l'avion. Cependant, seul l'ED-202A a été publié à ce jour.

Notre contribution dans ce contexte a été tout d'abord la définition des principales activités du futur processus de sécurité à partir des standards en construction. Une bonne partie de ce travail a été effectuée dans le contexte du projet SEISES[5] qui a mené à la définition du triple cycle en V qui montre ces activités et les interactions entre les processus de sécurité, de sûreté et de développement. L'autre tâche que nous avons effectuée dans ce domaine a été la définition d'une méthodologie d'analyse de risque intégrable aux prémices du cycle de développement d'un système. La méthode consiste à identifier les biens à protéger, les potentielles menaces et évaluer le risque de façon semi-quantitative afin de déployer les stratégies de mitigation adéquates. Le travail concernant cette partie a été publié à la conférence SafeComp en 2012.

1.2.2. EN TERMES DE DETECTION D'INTRUSIONS

L'intérêt de détecter l'exploitation de vulnérabilités à bord d'un avion en temps réel est de mener des actions afin d'éradiquer ou de réduire l'impact d'une attaque. Or, le problème de la plupart des systèmes de détection d'intrusion est que leur efficacité repose quasi-exclusivement sur l'exhaustivité de leur base de données de signatures d'attaques. Ce genre de système de détection requiert des mises à jour fréquentes et est dans l'incapacité de détecter des attaques dont on ignore la signature. Cependant, il y a un autre aspect à prendre en compte qui est la durée de vie d'un avion (environ 30 ans) donc l'idéal serait d'avoir des solutions de sécurité qui perdurent sur le long terme sans besoin de

[5] Systèmes Embarqués Informatisés, Sûrs et Sécurisés est un projet collaboratif d'Aerospace Valley entre Airbus, Rockwell Collins, Astrium, Serma Technologies, Apsys, EADS, Onera, DGA, Thales Avionics, LSTI, LAAS-CNRS pour la définition et l'interfaçage des processus de sécurité et de sûreté lors du design de systèmes embarqués.

mises à jour, i.e. génériques et autonomes. Il existe des techniques basées sur l'apprentissage qui utilisent des techniques de *Machine Learning* pour détecter des déviations comportementales. Cependant, ces techniques sont parfois compliquées à paramétrer et produisent souvent des fausses alertes, qui sont tout aussi dangereuses qu'une attaque.

Dans cette thèse, nous proposons une fonction d'audit de sécurité, générique et autonome, pour surveiller de façon continue les réseaux bord, basée sur des techniques de *Machine Learning*. L'objectif est de détecter dans un premier temps les attaques telles que la reconnaissance de réseau (*scans*) et les attaques par déni de service originées du côté monde ouvert et pouvant livrer des informations ou malmener les fonctions critiques du cœur avionique. Notons que dans le cas d'étude choisi pour cette thèse, le réseau avionique (nommé ci-après ADN pour *Aircraft Data Network*) est relié au réseau monde ouvert (nommé ci-après EON pour *Ethernet Open Network*) par une passerelle. Cette fonction est découpée en quatre étapes :

1. Capture des paquets
2. Caractérisation du trafic capturé à l'aide d'attributs descriptifs et prétraitement (mise à l'échelle des données, élimination des échantillons redondants)
3. Classification des échantillons : l'algorithme de Machine Learning reçoit les échantillons produits à l'étape 2 afin de déterminer son classement (i.e. échantillon « normal » ou « anormal »)
4. Post-traitement des résultats obtenus à l'étape 3 afin de réduire la quantité de faux positifs et caractériser l'attaque

Afin de détecter ces attaques, nous proposons différentes briques pour constituer les étapes ci-dessus: deux façons de modéliser le trafic bord au travers d'attributs pour l'étape 2, deux techniques de classification (l'une supervisée utilisant la théorie des Séparateurs à Vaste Marge et l'autre non-supervisée basée sur le *clustering* de sous-espace) pour l'étape 3. Enfin, pour se débarrasser des éventuels faux positifs (i.e. fausses alertes) nous proposons le calcul du *Local Outlier Factor* (un coefficient basé sur la densité afin de comparer sa densité à celle de ses voisins) à l'étape 4. Nos premiers résultats pour cette fonction d'audit ont été publiés à la conférence DASC (*Digital Avionics Systems Conference*) en 2013.

1.3. STRUCTURE DE LA THESE

La thèse est composée comme suit : le CHAPITRE 2 illustre le contexte industriel de cette thèse en listant quelques cyber-incidents mineurs qui ont touché l'environnement aéronautique, et les évolutions présentes et futures des systèmes embarqués qui peuvent les rendre vulnérables aux attaques. Après avoir défini les concepts de base de l'analyse de risqué, le CHAPITRE 3 décrit les différentes étapes de notre proposition de méthodologie d'analyse de risque et le travail réalisé concernant le processus de sécurité. Le CHAPITRE 4 dresse l'état de l'art sur les techniques de détection d'intrusion basées sur le *Machine Learning*. Ensuite, le CHAPITRE 5 donne la description fonctionnelle de notre fonction autonome de détection d'intrusions ainsi que l'évaluation de ses performances avec les différentes options présentées. Enfin, le CHAPITRE 6 conclut cette dissertation en résumant les principales contributions avec leurs avantages et inconvénients, des pistes d'amélioration et des perspectives.

CHAPITRE 2 :

CONTEXTE AERONAUTIQUE

Ce chapitre décrit le contexte industriel dans lequel s'est déroulé cette thèse, afin de justifier le besoin croissant d'analyser les risques de sécurité et celui d'introduire des contremesures dans les systèmes avioniques. Dans ce sens, on liste quelques cyber-attaques bénignes qui ont déjà été enregistrées dans le domaine de l'aéronautique ainsi que dans d'autres domaines où l'impact a été plus important. Ensuite, l'évolution des systèmes avioniques est survolée afin de voir à quel point les architectures deviennent interconnectées et donc vulnérables à des attaques. Enfin, ce chapitre décrit les standards de sécurité aéronautique et fait un bref état de l'art sur les solutions de sécurité embarquées en cours de recherche.

2.1. QUELQUES CYBER-INCIDENTS

Tout commença dans les années 60, lorsque le terme "hacker" apparut, à l'époque, ce mot désignait une personne capable de pousser les programmes informatiques au-delà de leurs fonctions d'origine. Depuis lors, les cyber-attaques ne cessent d'évoluer selon les enjeux : espionnage politique ou financier, ternissement de marque, terrorisme, etc. Bien que pour l'instant, aucune cyber-attaque n'ait été déclarée comme étant à l'origine d'un incident aéronautique catastrophique, un certain nombre d'attaques bénignes ont déjà eu lieu. Rien qu'en 2008, plus de 800 cyber-incidents ont été registrés par l'*Air Traffic Organization* (ATO) dont 17% ne furent pas résolus ! Ceci laisse présager que le piratage d'un avion n'est plus du ressort de la science-fiction mais plutôt une question de temps.

2.1.1. CYBER-ATTAQUES SUBIES DANS LE DOMAINE AERONAUTIQUE

2.1.1.1. ATTAQUES AU SOL

En 1997, un adolescent a réussi à s'introduire dans un ordinateur de Bell Atlantic, en provoquant la panne de tous les systèmes de communication de l'aéroport de Worcester (Massachusetts) et semant le chaos. En 2006, un virus informatique s'est répandu dans les systèmes de contrôle du trafic aérien ce qui obligea la fermeture d'une portion du trafic aérien de la FAA en Alaska. En 2007 un autre virus s'est répandu dans les EFBs (*Electronic Flight Bags*) de la flotte de Thaï Airways, il avait la capacité de rendre les EFB inutilisables. En 2009, le ver Downadup/Conficker a touché les réseaux de l'armée française, ce virus exploitait une vulnérabilité déjà connue de Windows Server Service. Les conséquences ont été que les plans de vol provenant des bases de données affectées ne pouvaient être chargés sur les avions de chasse. En 2011, encore un autre virus s'est répandu dans les systèmes de contrôle au sol des drones Predator et Reaper dans la base de Creech Air Force (Nevada).

2.1.1.2. Incidents Causes par une Mauvaise Utilisation

Pour avoir des conséquences néfastes, une attaque ne doit pas nécessairement être très élaborée. En effet, il y a eu des incidents originés par une mauvaise utilisation des outils ou bien des erreurs de saisie sur les ordinateurs de maintenance. Par exemple en 2006, un B747 dût faire un atterrissage d'urgence à l'aéroport de Paris-Orly après avoir décollé avec une vitesse anormalement basse, ce qui causa des dommages à sa gouverne. En fait, le co-pilote avait saisi la valeur du poids sans fuel (ou *Zero-Fuel Weight*) au lieu du poids au décollage (ou *Take-Off Weight*) sur l'ordinateur de bord. Un cas similaire arriva en 2004, lorsqu'un *MK Airlines 747 freighter* s'écrasa car les pilotes réutilisèrent les valeurs de répartition du poids du précédent vol au moment de calculer les vitesses et angles de décollage pour le prochain vol.

2.1.1.3. Quid du Piratage d'Avions?

Une attaque est souvent précédée d'une phase d'écoute du réseau afin de rassembler des informations sur ses échanges. Il est plutôt facile d'écouter les échanges ACARS (*Aircraft Communication Addressing and Reporting System*) ayant lieu entre l'avion et le sol. Il suffit de télécharger gratuitement le décodeur *acarsd*[6] et installer une antenne. En 2012, des portes dérobées (ou *backdoors*) furent trouvées dans des puces électroniques utilisées pour des applications militaires ainsi que dans des Boeing 787. Ces portes dérobées étaient délibérément installées dans le silicium pour faciliter les opérations de débogage et de réinitialisation de la mémoire. Cette puce pouvait donc être reprogrammée de sorte à laisser passer un très fort courant pouvant la brûler et la rendre inutilisable. Hugo Teso, un ancien pilote de ligne espagnol reconverti en expert sécurité créa une forte polémique à la conférence Hack in the Box à Amsterdam en avril 2013. Selon lui, il serait possible de prendre le contrôle d'un avion depuis le sol en utilisant son application android en exploit les vulnérabilités du protocole ACARS. Une attaque comme celle-ci pourrait être simplement contrée en désengageant le pilote automatique et en prenant les commandes de l'avion manuellement. Cependant, cette théorie est remise en question par la FAA (*Federal Aviation Administration*) étant donné que les tests de Teso furent menés sur des simulateurs de vol qui ne sont pas aussi robustes que les systèmes certifiés à bord des avions. Il a été néanmoins reconnu que les cyber-menaces doivent être maintenues sous étroite surveillance.

2.1.2. Attaques dans d'Autres Domaines

De nombreux chercheurs ont tiré la sonnette d'alarme concernant les vulnérabilités dans les systèmes qui composent les voitures récentes, qui peuvent être volées en utilisant des clés intelligentes, dans lesquelles il est possible d'inhabiliter le démarreur à travers le système télématique ou bien les freins au travers d'un malware mp3, ou encore contrôler n'importe quel système en installant un programme malveillant au travers de l'interface OBD-II (*On Board Diagnosis Interface*). La mort du journaliste américain Michael Hastings aurait été cause par le piratage de sa voiture. Dans un tout autre domaine, le ver *Stuxnet* fut développé par les Etats-Unis et Israël pour attaquer les usines d'enrichissement d'uranium en 2009. Celui-ci était capable de modifier la vitesse de rotation des turbines ce qui causa

[6] acarsd.org

d'importants dommages structuraux et ralentit la production d'uranium enrichi pendant plusieurs semaines. Il serait aussi possible d'ouvrir les portes des cellules de certaines prisons en utilisant des portes dérobées ou en exploitant des vulnérabilités dans leurs systèmes de commande.

Après des exemples aussi effrayants, l'on peut se demander : "pourquoi des avions n'ont-ils pas été encore cyber-attaqués ? ". La réponse est simple : jusqu'à présent, les avions étaient intrinsèquement sécurisés du point de vue réseau. Dans la suite, nous expliquons les évolutions récentes qui pourraient rendre les avions vulnérables aux cyber-attaques.

2.2. L'ÉVOLUTION DES SYSTEMES EMBARQUES

Auparavant, les systèmes critiques embarqués dans les avions étaient exclusivement dédiés à leur domaine et isolés de toute connexion avec l'extérieur, mais cette ségrégation tend à disparaitre comme le montrent les exemples qui suivent.

Domaine	AVANT	APRÈS
DES ARCHITECTURES FEDEREES AUX ARCHITECTURES INTEGREES	Les systèmes de commande d'un avion étaient composés de LRUs (*Line Replaceable Units*), i.e. des équipements associant matériel et logiciel pouvant être rapidement enlevés, remplacés et rebranchés pendant les opérations de maintenance. Dans les architectures fédérées, un LRU sert à une fonction donnée, occupe une place prédéfinie dans la baie avionique, est produite par un fournisseur donné et ce, spécifiquement pour un seul type d'avion. Le nombre de LRUs peut atteindre jusqu'à 20-30 unités reliés par plus de 100km de câbles !	Pour réduire le poids, la consommation d'énergie ainsi que les coûts de design et de maintenance, pour se libérer de la dépendance à un fournisseur d'équipements et optimiser le temps de maintenance au sol, les architectures intégrées furent introduites. Connues sous le nom d'IMA (*Integrated Modular Avionics*), permettent de n'embarquer que 6 à 8 calculateurs partitionnés de sorte à pouvoir héberger plusieurs fonctions. On peut donc renouveler la partie logicielle sans avoir à retirer la partie matérielle.
DE L'A429 A L'ADN (AIRCRAFT DATA NETWORK)	L'ARINC 429 est une norme qui décrit la topologie d'un réseau-bus pour l'aviation commerciale. Elle spécifie un protocole de transmission point-à-point unidirectionnelle au travers d'une paire torsadée (avec un ratio de 20 récepteurs pour un seul émetteur). Simple et déterministe, sans possibilité de collision, cette technologie est très fiable et sûre,	Ensuite, pour répondre aux besoins imposés par les architectures intégrées, l'ADN, aussi connu sous le nom d'AFDX (*Avionics Full-DupleX switched Ethernet*) fut introduit sur l'Airbus A380, le Boeing 787 Dreamliner et est spécifié dans la norme ARINC 664. Ce protocole Ethernet déterministe a permis de réduire considérablement le câblage

	mais une bande passante limitée (2 vitesses: high=100Ko/s et low=12,5Ko/s), et le poids du câblage est considérable. Pour améliorer ces points, l'ARINC 629 fut introduit avec un protocole pour partager un bus entre 128 émetteurs-récepteurs et une bande passante de 2 Mo/s.	en remplaçant les câblages point-à-point par des liens virtuels et d'atteindre une vitesse de 100Mo/s. Grâce à sa paire redondante de réseaux, l'ADN garantit la bande passante, la qualité de service (dont l'impossibilité de collision), etc.
DISTRIBUTION DE LOGICIEL	Dans les architectures fédérées, le logiciel était pré-chargé sur les LRUs. Le processus de distribution logicielle était indissociable de la matérielle et consistait à remplacer le LRU correspondant dans la baie avionique lors des phases de maintenance au sol.	Pour satisfaire les besoins des architectures intégrées, la distribution logicielle se fait au travers de supports tels que des disquettes, CD-ROMs scellés provenant des fournisseurs de logiciel. Dernièrement, d'autres options telles que le téléchargement de logiciel directement auprès des serveurs des fournisseurs, et installation en utilisant une clé USB ou pire, le wifi bord-sol de l'aéroport.
MODIFICATION DE LOGICIEL	La modification logicielle demandait un lourd processus de modification et éventuellement de re-certification.	Il existe maintenant des parties logicielles peu critiques qui peuvent être modifiées et/ou re-configurées par la compagnie aérienne elle-même sans besoin de re-certification c'est ce qu'on appelle les *User Modifiable Software*.
INTRODUCTION DE COMPOSANTS SUR ETAGERE	Sur les anciens modèles d'avion, tous les équipements à bord étaient dédiés à leur fonction.	Pour réduire les coûts de design et de production il a fallu opter pour plus de généricité et employer les composants sur étagère.
INTERNET A BORD	Pour éviter toute interférence, tous les appareils électroniques des passagers devaient être éteints notamment au décollage et à l'atterrissage.	Les nouveaux avions ont des bornes WiFi auxquelles les passagers peuvent se connecter depuis leurs PCs, tablettes ou téléphones portables.

DES ÉCHANGES VOCAUX AUX MESSAGES NUMERISES	Les communications bord-sol se font au travers d'échanges radio au niveau continental et Satellite dans les régions océaniques. Les communications radio ont certains inconvénients: la saturation de fréquence et la coordination de secteurs qui rend la tâche des contrôleurs aériens souvent compliquée.	Pour y remédier, le protocole ACARS (*Aircraft Communications Addressing and Reporting System*) fut introduit pour envoyer des messages numériques. Cependant, ces canaux bord-sol ne sont pas réservés uniquement aux opérations de navigation.
INTEROPERABILITE DES RESEAUX BORD-SOL ET HOMOGENEISATION DES PROTOCOLES		Pour permettre l'interopérabilité entre les réseaux bord, les composants sur étagère et les systèmes au sol, il est nécessaire d'homogénéiser les protocoles de communication, d'où l'introduction d'IPv4 et IPv6.).

Table 1 – Évolution des systèmes embarqués, comparaison avant-après

MENACES ASSOCIEES AUX ARCHITECTURES COMPLEXES ET INTERCONNECTEES

La connectivité et la complexité croissantes des systèmes embarqués incrémente les rend vulnérables à 4 principaux aspects :

- défaillances/pannes intrinsèques de composants ou de systèmes (sûreté),
- erreurs de design et de développement,
- fautes de mauvaise utilisation,
- attaques délibérées (sécurité).

Souvent les termes sûreté et sécurité mènent à confusion. La sûreté traite de la prévention des défaillances, alors que la sécurité s'occupe des attaques délibérées. Les erreurs de design et de développement sont traitées par les deux processus étant donné qu'elles peuvent être la source de pannes mais aussi de vulnérabilités ou de brèches de sécurité. Les problèmes de mauvaise utilisation sont considérés aussi bien par les facteurs humains et par le processus de sécurité.

CHAPITRE 3 : ANALYSE DE RISQUE

Les vulnérabilités suivent un cycle de découverte-exploit-révélation-patch, or, ceci n'est pas toujours compatible avec les pratiques en sûreté de fonctionnement. Le but des avionneurs est d'insérer le processus de sécurité au tout début du design d'une architecture. Si la sécurité est traitée une fois que les systèmes ont été implémentés, les coûts de redéveloppement et de re-certification sont très importants. Par ailleurs, le sur-design de contremesures doit être évité pour réduire des coûts de développement non nécessaires : le risque doit être quantifié pour ordonner ce qui doit être sécurisé en priorité et à quel niveau. Ce chapitre définit les notions et les étapes d'une méthodologie d'analyse de risque.

Le standard ED-203 doit fournir les méthodes d'analyse de risque, mais celui-ci est encore en cours de rédaction et donc inutilisable. En outre, les méthodologies d'analyse de risque existantes (EBIOS, MEHARI, ISO 27005, CRAMM, COBRA, etc.) pourraient convenir en les adaptant au monde aéronautique. Or, certains des outils et méthodologies requièrent la certification des analystes, ce qui est onéreux, d'où le besoin de créer une méthodologie simple et adaptable propre à l'aéronautique en attendant l'issue de l'ED-203. Notre contribution consiste en une méthode d'estimation du risque semi-quantitative basée sur des tables de caractérisation adaptables pour classer les risques.

3.1. DEFINITIONS ET CONCEPTS D'ANALYSE DU RISQUE

Assets. Dans la sécurité de l'information, un *asset* est un bien ou une ressource de valeur. Dans notre méthodologie, nous distinguons deux classes d'*assets* : les ***assets*** primaires (fonctions ou données critiques d'un avion) et les ***assets*** support (équipements avioniques qui brassent ces données ou exécutent ces fonctions) qui peuvent potentiellement avoir des vulnérabilités, i.e. des faiblesses exploitables par un attaquant qu'un attaquant peut exploiter afin de mettre un *asset* primaire dans une Condition de Menace, et que l'on va contrer en renforçant ces équipements par des contremesures.

Menace. Événement ou circonstance pouvant potentiellement affecter l'avion suite à une action humaine (intentionnelle ou non) résultant d'un accès non autorisé, une mauvaise utilisation, la modification, la destruction ou le déni d'accès au sein du système d'information de l'avion. Ces menaces sont créées par une source de menace qui a des motivations propres et différents moyens (vecteurs de menace) pour compromettre les *assets*. L'agent de menace peut être aussi bien un attaquant malintentionné qu'un utilisateur légitime qui commettrait une erreur d'utilisation. Le résultat d'une attaque réussie est une Condition de Menace qui affecte négativement la sûreté de vol (exemple: perte ou modification d'une fonction critique).

Contremesure. Renforce les vulnérabilités contre des attaques potentielles. Elle peut être soit *technique* (anti-virus, firewall, authentification, PKI) soit *organisationnelle* (procédures, restrictions d'accès). Les contremesures elles-mêmes doivent être considérées comme des *assets* support, on doit donc s'assurer

qu'elles n'interfèrent pas avec les activités opérationnelles ou la sûreté de fonctionnement et qu'elles n'introduisent pas des vulnérabilités additionnelles dans les systèmes.

Scenario de menace. Établir un scenario de menace consiste à identifier précisément CE QUI doit être protégé (*assets* primaire et support, leurs potentielles vulnérabilités, tout comme les contremesures déjà existantes), contre QUOI (caractériser le profil de l'attaquant) et POURQUOI (afin d'éviter ou réduire l'impact d'une Condition de Menace).

Impact. L'impact d'un scenario de menace est la conséquence d'une attaque réussie. Dans une analyse de risqué classique, l'impact est souvent mesuré en termes de pertes économiques, dommage à l'image d'une personne ou à une marque, révélation d'informations confidentielles, violation de privacité, etc. En aéronautique, le seul aspect considéré est la sécurité pour la sûreté (***security for safety***). L'impact est uniquement lié aux événements craints qui peuvent affecter la sûreté de vol : en termes de conséquences sur les capacités fonctionnelles de l'avion, la surcharge de travail des pilotes et l'effet sur les passagers comme décrit dans la table 2.

Risque. Dans la littérature, le risque est communément défini comme le produit des facteurs : *Risque = Menace × Vulnérabilité × Impact.* Dans notre cas, on définit le risque comme la combinaison de l'impact d'une attaque réussie et la vraisemblance d'une telle attaque. La vraisemblance est elle-même définie dans notre méthode comme la combinaison de la capacité de l'attaquant et l'exposition de l'*asset*.

Sévérité	Effet des conditions de panne sur :			Probabilité	DAL
	Fonctions avioniques	Charge de travail du personnel à bord	Passagers		
Aucun effet	Aucun effet			Fréquent = 1	E
Mineure	Légère réduction	Légère augmentation	Inconvénients ressentis	Probable = 10^{-3}	D
Majeure	Réduction significative	Augmentation significative, conditions qui réduisent l'efficacité du personnel	Certain inconfort	Rare = 10^{-5}	C
Hasardeuse	Importante réduction	Importante charge de travail ou gênes physiques qui font que le personnel n'est plus en mesure d'accomplir ses tâches complètement	Effets adverses	Extrêmement rare = 10^{-7}	B
Catastrophique	Les conditions de panne empêchent l'assurance d'un vol et/ou d'un atterrissage sûr			Extrêmement improbable = 10^{-9}	A

Table 2 – Niveaux de sévérité considérés en sûreté de fonctionnement aéronautique (tirée du standard SAE ARP-4761)

3.2. DESCRIPTION DE LA METHODOLOGIE D'ANALYSE DE RISQUE

Cette partie décrit les six étapes de notre méthodologie d'analyse de risque, résumés sur la Figure 1. Cette méthodologie se veut être simple, adaptable au contexte aéronautique et répondant aux besoins des standards de l'ED-202A et ED-203 en cours de construction. Elle à modéliser le contexte d'analyse afin d'identifier les scénarios de menace en utilisant une approche duale (*top-down* et *bottom-up*). Ces scénarios sont ensuite évalués : leur vraisemblance, leur impact et l'acceptabilité du risque. Si le risque n'est pas acceptable, les spécifications pour les contremesures doivent être formalisées et associées à un niveau de sécurité ou SL (pour *Security Level*) pour indiquer l'effort pour contrer cette menace. Finalement, les contremesures sont implémentées et le système final doit être réévalué à nouveau.

3.2.1. ÉTAPE 1 : ÉTABLISSEMENT DU CONTEXTE

Avant toute analyse, il faut collecter toute l'information nécessaire et la modéliser pour avoir une vision précise du périmètre de sécurité à analyser : les interfaces et interactions fonctionnelles, les conditions d'utilisation des systèmes, les architectures, les standards à être appliqués, le cadre légal, etc.

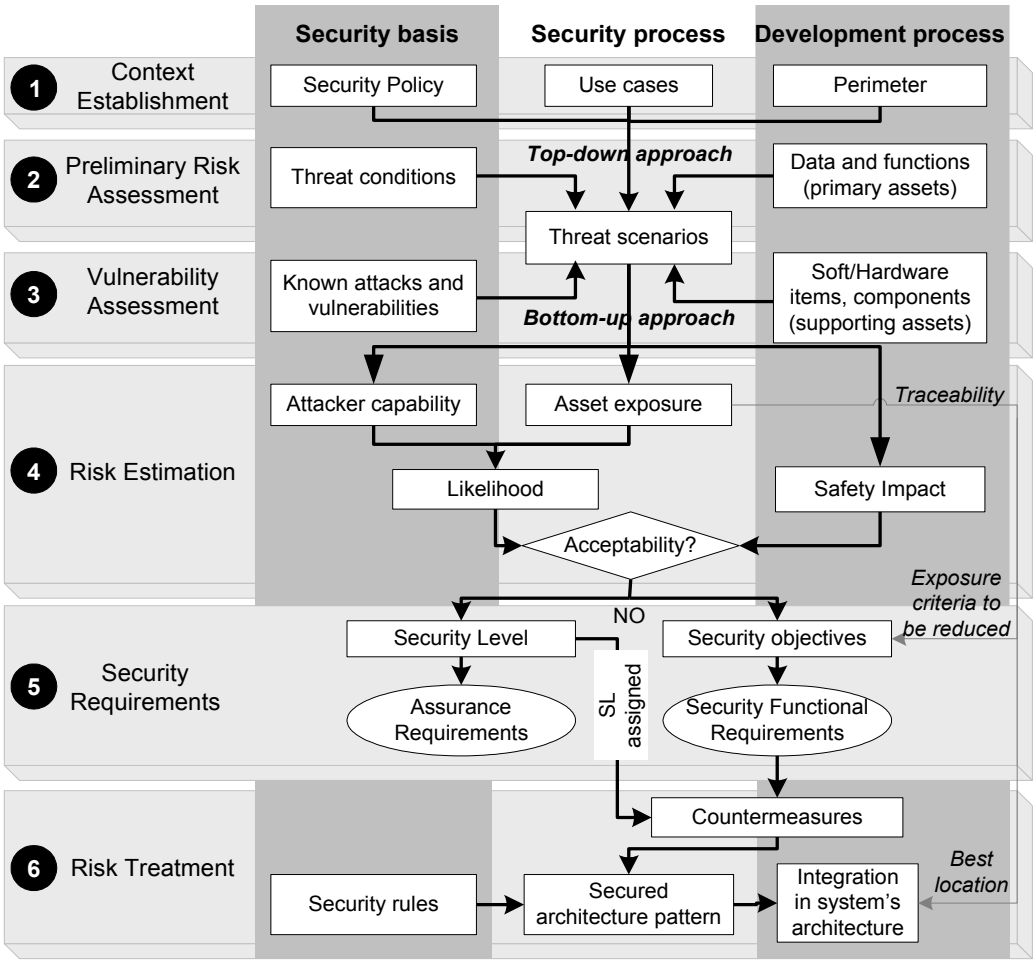


Figure 1 – Processus d'analyse de risque proposé

3.2.2. ÉTAPE 2 : ANALYSE PRELIMINAIRE (APPROCHE TOP-DOWN)

L'analyse de risqué préliminaire est une activité se faisant dans les premières étapes du design d'un système (i.e. c'est une analyse fonctionnelle pour laquelle on n'a pas forcément besoin de l'architecture détaillée). Le but est d'aider les concepteurs à prendre en compte les principaux enjeux de sécurité au moment de définir l'architecture d'une suite avionique. Il s'agit d'identifier ce qui doit être protégé (i.e. les *assets* primaires) et les événements craints (i.e. les conditions de menace).

Identification des *assets* primaires et des conditions de menace. Tout d'abord, les fonctions critiques sont identifiées depuis la FHA (*Functional Hazard Assessment*). La FHA, basée sur l'ARP-4761, est une analyse de sûreté de fonctionnement faite à haut niveau et assez tôt dans le développement d'un système, elle consiste à classifier les conditions de panne applicables aux fonctions avioniques en fonction de leur gravité.

Approche *top-down* pour la définition des scenarios de menace. A partir d'un événement craint (i.e. une condition de menace) sur un *asset* primaire, toutes les attaques ou mauvaises utilisations pouvant les causer sont considérées en les déduisant au niveau système-de-systèmes (avion ou suite avionique), puis au niveau système et sous-système.

3.2.3. ÉTAPE 3 : ANALYSE DE VULNERABILITE (APPROCHE BOTTOM-UP)

Une fois l'architecture définie et les choix d'implémentation connus, tous les *assets* supports associés à chacun des *assets* primaires sont identifiés. Il faut vérifier si ces *assets* support possèdent déjà des contremesures de sécurité et surtout s'ils contiennent d'entrée de jeu des vulnérabilités connues. Pour cela, on prône l'utilisation d'une checklist générique de vulnérabilités comme la base de données CVE[7] (*Common Vulnerabilities and Exposures*) pour les composants sur étagère et des tests d'intrusion pour les équipements dédiés. Le but de cette étape est d'identifier les potentielles vulnérabilités sur les *assets* support au niveau composant et les failles au niveau des interfaces homme-machine et entre systèmes afin de déduire les conséquences d'une exploitation de ces vulnérabilités au niveau système et système-de-systèmes.

Complémentarité des approches *top-down* et *bottom-up* pour la définition des scenarios de menace. Pour résumer, l'approche *top-down* permet l'identification des besoins de sécurité fonctionnels de haut-niveau, tandis que l'approche *bottom-up* permet de valider et compléter ces besoins avec les contraintes d'implémentation causées par les choix techniques. Elles sont donc complémentaires.

3.2.4. ÉTAPE 4 : ESTIMATION DU RISQUE

Puisqu'il est impossible de traiter tous les scenarios de menace identifiés, il est nécessaire de les classer par ordre de criticité. L'on définit le risqué d'un scenario de menace comme la combinaison de sa vraisemblance (ou *likelihood*) et son impact sur la sûreté de vol (*safety impact*). Le *likelihood* lui-même est défini par la combinaison des facteurs : capacité de l'attaquant et exposition de *l'asset*. Voici la méthode pour l'évaluation semi-quantitative de ces critères :

[7] http://cve.mitre.org/

Soit $X = \{X_1, \dots, X_n\}$ un set de n attributs qualitatifs choisis pour caractériser la capacité d'un attaquant requise pour mener une attaque et de façon similaire $Y = \{Y_1, \dots, Y_z\}$ un set de z attributs qualitatifs choisis pour caractériser l'exposition de l'*asset*. Chaque attribut X_i peut prendre m valeurs : $\{X_i^1, \dots, X_i^m\}$, X_i^j étant plus critique que X_i^{j-1}. A chaque valeur qualitative X_i^j, on associe des degrés quantitatifs de criticité x_i^j, avec $x_i^j > x_i^{j-1}$.

Par exemple :

$X = \{X_1 =$ "temps requis pour mener l'attaque",
$X_2 =$ "expertise de l'attaquant",
$X_3 =$ "connaissance préalable du système",
$X_4 =$ "équipement utilisé",
$X_5 =$ "localisation de l'attaquant"$\}$.

X_1 peut prendre les valeurs :
{="> jour", → x_1^1=0
="< jour", → x_1^2=1
="heures", → x_1^3=2
="minutes"} → x_1^4=3

Appelons $f_j()$ la fonction d'évaluation faite par l'analyse de sécurité afin d'assigner le degré de criticité correspondant a_i pour chaque attribut X_i pour un scénario de menace donné : $a_i = f_{j=1}^{m}(x_i^j)$. La capacité de l'attaquant est exprimé pour chaque scenario de menace comme la somme normalisée des valeurs assignées à tous les attributs du set X (cf. équation 1). Le même raisonnement est utilisé pour déterminer le degré d'exposition d'un *asset*.

$$A = \frac{\sum_{i=1}^{n}(\mathrm{a_i})}{\sum_{i=1}^{n}(\mathrm{x_i^m})}, \quad \mathrm{x_i^m} \geq \mathrm{x_i^j}, \forall \mathrm{i} = 1 \dots \mathrm{n}, \forall \mathrm{j} = 1 \dots \mathrm{m} \qquad (1)$$

Nous allons à présent donner des exemples d'évaluation de la capacité de l'attaquant et de l'exposition de *l'asset*. L'objectif est que selon le contexte d'analyse, les analystes de sécurité définissent le nombre et le type d'attributs ainsi que des valeurs adéquats.

Capacité de l'attaquant. Pour déterminer les valeurs de sévérité, l'on raisonne en termes de fréquence: plus une attaque risqué de se produire souvent et plus la capacité de l'attaquant sera importante. Par exemple, il est plus vraisemblable d'avoir des attaques fréquentes de la part de *script kiddies* (i.e. des personnes qui n'ont pas de réelles capacités de hacker mais qui tentent de s'infiltrer dans les systèmes par amusement) que de la part d'organisations criminelles/terroristes. Un exemple est donné table 3.

	Valeurs			
Attributs	3	2	1	0
X1: Temps requis pour mener l'attaque	minutes	heures	< jour	> jour
X2: Expertise de l'attaquant	employé	amateur	avancé	expert
X3: Connaissance préalable du système	public	restreint	sensible	critique
X4: Équipement utilisé	aucun	domestique	spécialisé	dédié
X5: Localisation de l'attaquant	hors-aéroport	aéroport	cabine	cockpit

Table 3 – Exemple d'attributs et de valeurs pour la capacité de l'attaquant

Exposition de *l'asset*. En suivant le même principe, on construit une table (e.g. table 4) afin de mesurer jusqu'à quel point un *asset* est exposé (i.e. accessible) aux attaques. Pour cette table l'on raisonne cette fois-ci en termes de restriction d'accès : plus l'accès à un *asset* est restreint (physiquement en termes de connaissance de ses vulnérabilités) et plus la criticité et donc l'exposition de l'*asset* seront moindres.

	Valeurs				
Attributs	4	3	2	1	0
Y1: Localisation de l'*asset*	extérieur de l'avion	cabine	local de maintenance	cockpit	baie avionique
Y2: Classe[8] de l'*asset*	classe 1		classe 2		classe 3
Y3: DAL	DAL E	DAL D	DAL C	DAL B	DAL A
Y4: Vulnérabilités	connues du public large	connues du public limité	non publiques	inconnues	aucune
Y5: Contremesures	aucune	organisationnelle	technique	sur l'*asset*	>2 protections

Table 4 – Exemple d'attributs et de valeurs pour l'exposition de l'*asset*

Vraisemblance. Il s'agit de l'estimation qualitative de la potentielle fréquence d'occurrence d'un scénario de menace. Le standard ED-202 considérait cinq niveaux de vraisemblance : 'pV: fréquent', 'pIV: probable', 'pIII: rare', 'pII: extrêmement rare', 'pI: extrêmement improbable' mais ne donne aucune information sur comment déterminer et justifier ces niveaux. Comme ils sont trop subjectifs pour être déterminés directement, nous utilisons la table 5 pour assigner un niveau de vraisemblance par la combinaison des valeurs semi-quantitatives de la capacité de l'attaquant (A) et de l'exposition de l'*asset* aux menaces (E). Notons que la table 5 est utilisable quel que soit le nombre d'attributs de de valeurs utilisés dans les tableaux précédents pour une évaluation plus flexible. Cependant, il faut soigner la taxonomie de ces critères afin que l'évaluation soit exhaustive, non ambiguë et répétable.

		Capacité de l'attaquant				
		$0 \le A \le 0,2$	$0,2 < A \le 0,4$	$0,4 < A \le 0,6$	$0,6 < A \le 0,8$	$0,8 < A \le 1$
Exposition de l'*asset*	$0 \le E \le 0,2$	pI	pI	pII	pIII	pIV
	$0,2 < E \le 0,4$	pI	pI	pII	pIII	pIV
	$0,4 < E \le 0,6$	pII	pII	pIII	pIV	pV
	$0,6 < E \le 0,8$	pIII	pIII	pIV	pV	pV
	$0,8 < E \le 1$	pIV	pIV	pV	pV	pV

Table 5 – Vraisemblance d'une attaque par la combinaison des caractéristiques de l'attaquant et l'exposition de l'*asset*

Acceptabilité du risque. Pour déterminer si un risque est acceptable ou non, et mesurer l'effort à être fourni pour éviter les conditions de menace les plus vraisemblables et dangereuses, l'on propose la matrice d'acceptabilité du risque suivante (table 6) qui associe l'impact *safety* (cf. table 2) et la vraisemblance (*likelihood*, cf. table 5).

		Impact *Safety*				
		Aucun effet	Mineur	Majeur	Hasardeux	Catastrophique
Vraisemblance	pV: Fréquent	Acceptable	Non acceptable	Non acceptable	Non acceptable	Non acceptable
	pIV: Probable	Acceptable	Acceptable	Non acceptable	Non acceptable	Non acceptable
	pIII: Rare	Acceptable	Acceptable	Acceptable	Non acceptable	Non acceptable
	pII: Extrêmement rare	Acceptable	Acceptable	Acceptable	Acceptable	Non acceptable
	pI: Extrêmement improbable	Acceptable	Acceptable	Acceptable	Acceptable	Acceptable*

* = il faut assurer qu'aucune vulnérabilité simple, attaquée de façon réussie, résulterait en une condition catastrophique

Table 6 – Matrice d'acceptabilité du risque

[8] Classe 1: Portable Electronic Device (PED); classe 2: PED modifié; classe 3: équipement développé et installé sous contrôle.

3.2.5. ÉTAPE 5 : SPECIFICATIONS DE SECURITE

Une fois les scenarios de menace établis et l'acceptabilité du risque évaluée, il faut déterminer le niveau de sécurité (ou *security level*):

***Security Level* (SL).** Le SL est similaire au *Design Assurance Level* (DAL) utilise dans le processus *safety* et défini dans la DO-178B. Ce niveau a une signification duale, il représente :

- l'efficacité des contremesures (il faut assurer que les fonctions de sécurité agissent correctement et de façon sûre)
- l'assurance d'une implémentation adéquate (il faut assurer que les contremesures de sécurité suivent un processus rigoureux de design et d'implémentation)

Pour chaque scénario de menace non acceptable identifié, un SL est déterminé en fonction du nombre de niveaux à réduire afin que le risque devienne acceptable dans la table 6. Selon si la vraisemblance doit être réduite de 0, 1, 2, 3 ou 4 niveaux pour atteindre un niveau acceptable, le SL va respectivement les valeurs E, D, C, B ou A. Un SL est assigné à chaque contremesure développée, les règles pour une implémentation adéquate seront données par le standard ED-203, une fois publié.

3.2.6. ÉTAPE 6 : TRAITEMENT DU RISQUE

Traitement du risque. Basé sur l'ISO 27005:2011, l'ED-202A propose les options suivantes pour le traitement des risques :

- Éviter le risque en modifiant le système de sorte à ce qu'il ne soit plus exposé à ce genre de risque
- Mitiger le risque en ajoutant des contremesures de sécurité afin de réduire l'impact et/ou la vraisemblance d'une potentielle attaque
- Accepter les conséquences d'une attaque sans implémenter de contremesures
- Transférer (ou partager) la responsabilité de traiter le risque avec d'autres organisations

CHAPITRE 4 :
THEORIE SUR L'AUDIT DE SECURITE

Ce chapitre vise à donner une vue d'ensemble sur les principes et algorithmes utilisés dans les systèmes de détection d'intrusions ou pour la détection d'anomalies. Après avoir défini ce qu'est un système de détection d'intrusions, nous décrirons les deux principales techniques de classification utilisées par les systèmes d'apprentissage (i.e. *Machine Learning*) qui sont l'apprentissage supervisé et non supervisé.

4.1. DIFFERENTS TYPES DE DETECTION D'INTRUSION: BASE SIGNATURE OU COMPORTEMENTAL

La plupart des systèmes de détection d'intrusion ou IDS (pour *Intrusion Detection Systems*) que l'on peut trouver dans le commerce utilisent des techniques basées sur la connaissance de la signature des attaques, i.e. ils requièrent donc une base de données assez fournie d'attaques déjà rencontrées ou un set de règles préétablies pour effectuer des comparaisons et lancer une alarme si l'une de ces signatures est rencontrée (comme le font la plupart des anti-virus) et/ou bloquer le trafic correspondant à une règle non respectée (comme sur les firewalls). Ce genre d'IDS présente l'avantage d'être très précis, ils génèrent très peu de fausses alertes et l'on peut rapidement connaître la nature de l'attaque et agir rapidement en conséquence. Cependant, ces contremesures ont vu leurs performances s'amoindrir à cause de la sophistication des nouvelles attaques, suffisamment élaborées pour les contourner (par exemples les virus polymorphiques). Les inconvénients de ces techniques basées sur les signatures sont nombreux : ils requièrent une mise à jour fréquente de leur base de données d'attaques, ce qui est long et coûteux à obtenir, et, de plus, elles ne permettent pas la détection de nouvelles attaques, inconnues à ce jour.

C'est pourquoi des IDS basés sur l'étude comportementale (ou ADS pour *Anomaly-based IDS*) de ce qui est normal ont été introduits afin de mener des mesures statistiques au sein du trafic normal pour obtenir un modèle comportemental en utilisant souvent des techniques de *Data Mining*. Les ADS consistent à observer toute déviation du modèle comportemental attendu d'un réseau en régime normal. Ils présentent l'avantage claire de permettre la détection de nouvelles attaques et ce de façon générique et autonome car ils ne requièrent pas de signatures. Cependant, les ADS comportent un haut risque de fausses alertes car, d'une part, l'apprentissage est difficilement exhaustif et d'autre part, les algorithmes de Machine Learning utilisés par ces ADS présentent des limites.

4.2. ALGORITHMES DE MACHINE LEARNING ET TECHNIQUES DE DETECTION D'ANOMALIES ASSOCIEES

Les hypothèses basiques formulées dans la détection d'anomalies en cyber-sécurité sont que d'une part, les attaques diffèrent forcément du trafic normal, et, d'autre part, les attaques sont rares et donc

représentent une très petite proportion du trafic. En classification, la recherche d'événements rares est aussi appelé recherche d'*outliers*. En *Machine Learning*, il y a deux principales techniques de classification les supervisées et les non-supervisées. Les techniques **supervisées** nécessitent une phase d'apprentissage avec un set de données bien connu et labélisé au préalable, tandis que celles **non-supervisées** ne requièrent aucune connaissance a priori du label des échantillons. Beaucoup d'entre eux ont été adaptés à la reconnaissance d'*outliers*.

4.2.1. APPRENTISSAGE SUPERVISE

Le but de l'apprentissage supervisé est de trouver une fonction ou un modèle qui à partir d'échantillons en entrée permet de prédire le plus précisément possible le label / nature / classe de cet échantillon. L'étape d'apprentissage est cruciale car les données pour l'apprentissage doivent être suffisamment nombreuses et représentatives pour que le modèle soit le plus exhaustif et précis possible, et l'apprentissage lui-même doit être suffisamment régulier pour prendre en compte les éventuelles fluctuations du trafic dans le temps. Les algorithmes purement supervisés tels que les arbres de décision, k-plus proches voisins, Séparateurs à Vaste Marge (SVM), Perceptron, Naïve Bayes, etc. sont rarement utilisés pour la détection d'intrusions car, en pratique, il est compliqué et coûteux de produire des labels sur un set de données. De plus, nous n'avons aucune connaissance a priori sur le type d'attaques, il est donc difficile de faire un apprentissage exhaustif. Cependant, ces algorithmes ont inspiré un certain nombre d'algorithmes semi-supervisés qui sont adaptés à la détection d'*outliers*.

4.2.2. APPRENTISSAGE NON-SUPERVISE

Le but de l'apprentissage non-supervisé est de découvrir des classes au sein des échantillons en les regroupant par similarité sans aucune connaissance préalable. Il est très souvent utilisé pour la classification du trafic de masse d'Internet et la détection d'anomalies car il est souvent compliqué d'assurer que le set d'apprentissage pour un algorithme supervisé est garanti 100% sans anomalie. Aussi, le trafic Internet varie beaucoup en volume, avec des fluctuations selon le jour/l'heure, l'évolution des topologies, l'apparition de nouveaux protocoles, la mutation des anomalies, etc., ce qui fait qu'un apprentissage sans aucun a priori est plus adapté. Entre autres techniques, le non-supervisé est très souvent associé aux techniques de *clustering*, très largement utilisées.

4.2.3. METHODES DE CLASSIFICATION DE CLASSE UNIQUE

Dans le cas particulier où nous réussissons à n'utiliser que des échantillons d'une seule classe, dans notre cas de label normal, on peut parler de classification de classe unique (ou *one-class*). Il y a beaucoup de méthodes statistiques pour la détection d'*outliers* en délimitant l'unique classe par des seuils et considérant tout ce qui y est extérieur comme des *outliers*. Dans cette thèse, nous avons décidé de nous concentrer sur l'One-Class SVM (OCSVM) qui reprend la technique SVM mais pour une classe unique.

CHAPITRE 5 : FONCTION DE DÉTECTION D'INTRUSION AUTONOME POUR LES RÉSEAUX AVIONIQUES

Ce chapitre décrit notre système d'audit des réseaux bord générique et autonome pour la détection d'intrusions en utilisant des algorithmes de Machine Learning supervisés et non-supervisés. Nous commençons par la description fonctionnelle de ce système d'audit, ensuite nous décrivons le contexte de son étude et de son évaluation.

5.1. PRINCIPE DE LA FONCTION D'AUDIT SECURITE DES RESEAUX BORD

La fonction d'audit de sécurité se compose d'une séquence de quatre étapes illustrées dans la figure 5.1 et décrites ci-après. Elle a été entièrement codée en Python en utilisant les librairies *Scapy*[9], pour la capture, la modification et l'analyse des champs des paquets et scikits-learn[10] (étapes 1 et 2), pour le test de différents algorithmes de *machine learning* (étape 3).

ÉTAPE 1: ACQUISITION DES PAQUETS. Cette première étape consiste à capturer de façon continue le trafic à des endroits critiques du réseau, i.e. au niveau des interfaces ou passerelles, et à dater les paquets capturés pour une meilleure traçabilité.

ÉTAPE 2: PRETRAITEMENT DES PAQUETS. Ensuite, les paquets sont groupés en échantillons dans des fenêtres d'observation de durée ΔT. Pour chacun de ces échantillons, l'on construit des attributs les caractérisant. Les attributs d'un échantillon $\boldsymbol{i}$ correspondent au vecteur $\boldsymbol{x} = \{\boldsymbol{x_i^1}, \boldsymbol{x_i^2}, \dots, \boldsymbol{x_i^N}\}$ des N principales caractéristiques qui permettent de modéliser le comportement du trafic normal du réseau, i.e. celles qui sont les plus à même de changer en cas d'attaque. Choisir des attributs adéquats est un réel challenge et un pilier fondamental pour une bonne détection.

ÉTAPE 3: CLASSIFICATION DES ÉCHANTILLONS. Dans cette étape, le but est de déterminer si les échantillons sont similaires aux autres occurrences et taguées comme normales (label +1) ou s'ils diffèrent du reste et donc taguées comme anormales (label -1). Pour cette étape, nous proposons deux options :

- Une technique non supervisée de *clustering* de sous-espace : puisque les performances des algorithmes de *clustering* ne sont pas optimales dans des espaces de grandes dimensions, le but

[9] http://www.secdev.org/projects/scapy/
[10] http://scikit-learn.org/

du *clustering* de sous-espace est de diviser l'espace des attributs en plusieurs sous-espaces de petite dimension, appliquer l'algorithme dans chacun d'entre eux et ensuite corréler les résultats.

- Une technique supervisée : l'One *Class Support Vector Machine* (OCSVM), l'algorithme apprend à modéliser un modèle de trafic normal au travers des attributs décrivant des échantillons d'entrainement exclusivement sans anomalie. Le modèle ainsi obtenu est alors gardé en mémoire de sorte à ce que l'algorithme soit capable de prédire si un nouvel échantillon capturé est inclus dans le modèle ou non, produisant un label pour chaque échantillon.

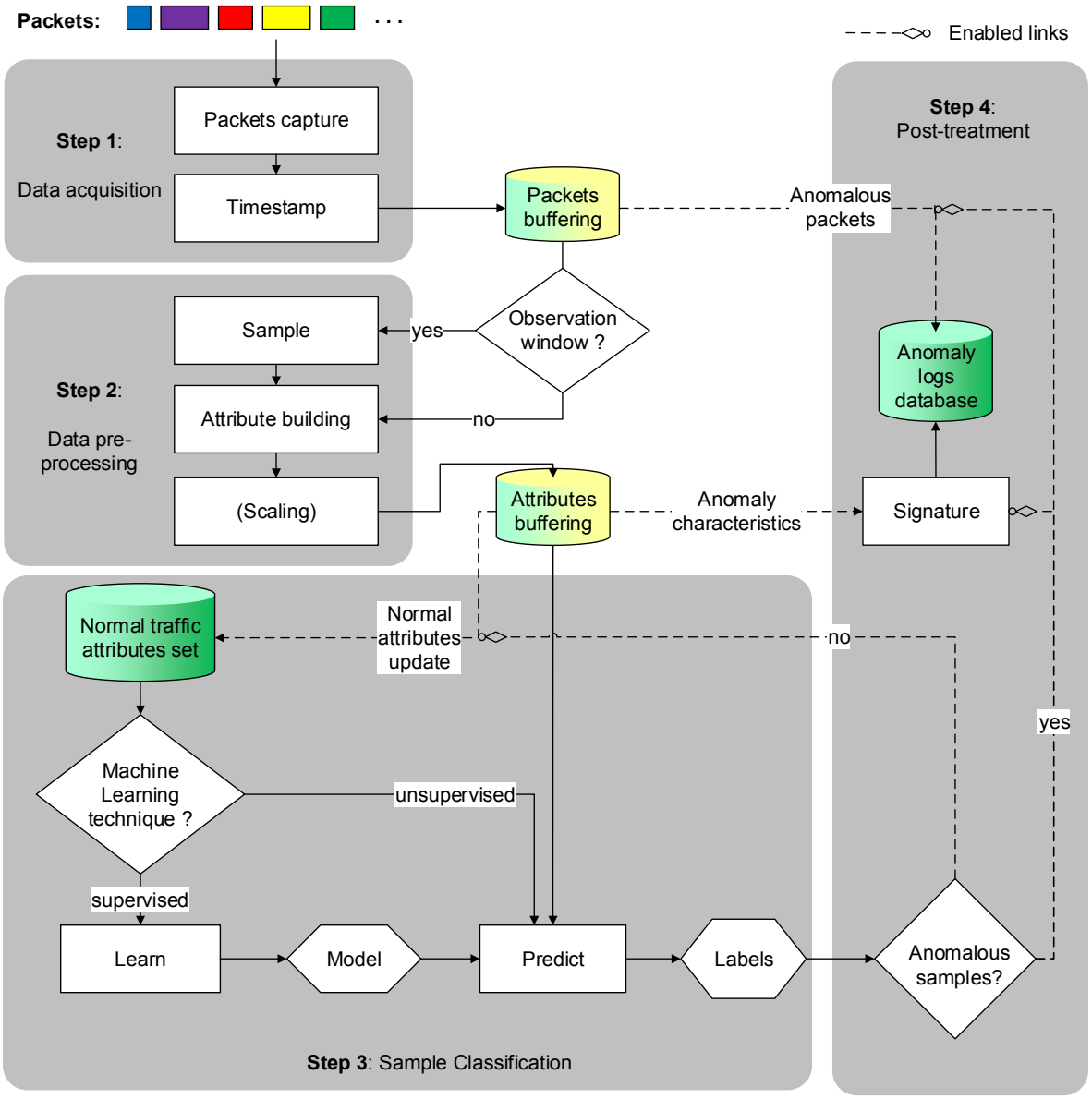


Figure 2 – Étapes de la fonction d'audit sécurité des réseaux bord

Le but de cette étape est de trouver la limite entre les échantillons normaux et anormaux. Or, il est possible que cette frontière ne soit pas clairement définie. En effet, selon les configurations utilisées dans les algorithmes de Machine Learning il est possible de surestimer ou sous-estimer le modèle. Ici, nous préférons avoir des fausses alertes plutôt que des faux négatifs. Pour distinguer des Faux Positifs

(fausses alertes) des Vrais Positifs (vraies anomalies), nous calculons le *Local Outlier Factor* pour chaque potentielle anomalie pour comparer sa densité locale à la densité locale de ses k plus proches voisins.

ÉTAPE 4: POST-TRAITEMENT. Une fois une anomalie a été détectée, l'on peut essayer de trouver sa signature en identifiant l'attribut pour lequel la distance entre l'anomalie et la frontière entre les échantillons normaux et anormaux est le plus significative. L'échantillon de paquets ainsi que la signature obtenue sont enregistrés pour une ultérieure référence ou éventuellement pour mener une action. Le buffer circulaire contenant les paquets peut être vidé des paquets considérés comme normaux. Aussi la base de données d'attributs normaux peut être renouvelée périodiquement au lieu d'en utiliser une de statique, pour que l'algorithme supervisé puisse prendre en compte les fluctuations possibles du trafic du monde ouvert.

5.2. CONTEXTE DE CAPTURE DU TRAFIC

5.2.1. SETS DE DONNEES

Les traces utilisées pour les tests ont été sniffées avec Wireshark au travers d'un hub au niveau de la passerelle entre le monde ouvert (i.e. entre les réseaux dédiés à la compagnie aérienne et aux passagers) et le réseau avionique critique.

Set d'apprentissage. Puisque les algorithmes de détection d'anomalie supervisés nécessitent d'être entraînés exclusivement avec des traces normales abondantes. Nous avons fait ces captures de part et d'autre de la passerelle et pour chacune des phases de vol pour qu'il soit le plus représentatif possible.

Set de test. Ce set, nécessaire pour évaluer la performance de prédiction de notre fonction d'audit, nous avons fait des captures Wireshark alors que des tests d'intrusion étaient menés au niveau de la passerelle, depuis le monde ouvert vers le réseau avionique. Toutes les attaques testées et décrites ci-après, ont eu lieu pendant les opérations de maintenance, i.e. pendant que l'avion est au sol.

5.2.2. PRINCIPALES CARACTERISTIQUES DU TRAFIC "NORMAL"

Détecter des intrusions sur Internet est bien plus compliqué que dans les réseaux avioniques. Sur Internet, le volume du trafic ne cesse de croître et de varier selon la saison, le jour de la semaine ou l'heure. Aussi, les systèmes de détection d'anomalies doivent affronter l'apparition de nouvelles applications et protocoles, ou bien des connexions massives à certains sites. Contrairement à l'évolution constante du trafic Internet, le comportement du trafic à bord des avions est déterministe.

5.2.3. ATTAQUES CONSIDEREES DANS LES TRACES CAPTUREES

Nom	Description

SCAN ARP	Envoi massif de requêtes ARP-"*who has*" à un certain nombre d'adresses IP afin d'identifier les adresses IP et MAC des différents hôtes disponibles dans le réseau.
SCAN UDP	Consiste à scanner une vaste plage de ports pour un hôte donné en envoyant des datagrammes UDP vides. S'il s'agit d'un port fermé l'on reçoit un message : "*Port Unreachable*", si l'on ne reçoit pas de message, il est probable qu'il s'agisse d'un port UDP ouvert.
SNMP BRUTE FORCE	SNMP (pour *Simple Network Management Protocol*) est un protocole pour aider les administrateurs réseau à gérer les appareils connectés à un réseau. Un scan SNMP sert à reconnaître si un nœud possède des propriétés SNMP, si celui-ci est disponible, ainsi que d'autres informations sur l'hôte.
FUZ-ZING	Consiste à envoyer des paquets modifiés aléatoirement à un hôte donné pour observer le comportement du réseau et/ou de l'hôte et y trouver des vulnérabilités.
TEAR-DROP	Consiste à envoyer un nombre important de paquets fragmentés dont l'offset se chevauche à un hôte donné pour le saturer (attaque de Déni de Service).
Rejeu de paquets	Modification de certains champs d'un paquet de façon consistante pour observer le comportement de l'hôte visé.

Table 7 – Attaques continues dans nos sets de données et description

5.3. METRIQUES D'ÉVALUATION DE PERFORMANCES

En Machine Learning, l'évaluation de la qualité de la classification évaluation est faite avec les quatre éléments de base qui sont : les vrais positifs (VP), i.e. les anomalies effectivement détectées, les faux négatifs (FN), i.e. les anomalies non-détectées, les faux positifs (FP), i.e. les fausses alarmes et les vrais négatifs (VN). On peut les résumer sous la forme d'une matrice de confusion (Table 8).

	Prédit Positif	Prédit Négatif
Vrai Positif	VP	FN
Vrai Négatif	FP	VN

Table 8 – Matrice de confusion pour un problème de classification à 2 classes

Pour mesurer le pourcentage d'échantillons correctement classés, on utilise l'incertitude (1) (i.e. le ratio de vrais positifs et vrais négatifs sur le nombre total d'échantillons). Cependant, il peut fausser l'évaluation lorsque le nombre de vrais négatifs est considérablement supérieur au nombre de vrais positifs. C'est pourquoi l'on utilise d'autres métriques en complément.

$$\text{Incertitude: I} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FN} + \text{FP} + \text{VN}} \quad (1)$$

Le premier critère est la précision (2), i.e. le ratio entre le nombre de varies anomalies et le nombre d'échantillons classés par l'algorithme comme des anomalies. Plus la valeur est petite et plus il y a de faux positifs. Le second critère est le rappel (3), qui est le pourcentage d'anomalies correctement

classées en tant que telles parmi toutes les vraies anomalies. Ainsi, plus le rappel est petit et plus il y a d'anomalies non détectées.

$$\text{Précision} : P = \frac{VP}{VP + FP} \quad (2) \qquad \text{Rappel} : R = \frac{VP}{VP + FN} \quad (3)$$

Souvent on utilise le F-mesure (4) qui est la moyenne harmonique de la précision et du rappel.

$$\text{Fmesure} : F = \frac{2 \times P \times R}{P + R} = \frac{2VP^2}{2VP^2 + VP(FN + FP)} \quad (4)$$

5.4. PROPOSITION D'ATTRIBUTS POUR L'ÉTAPE 2

Ce set d'attributs se construit pour caractériser chacun des échantillons de durée ΔT.

Num	Nom de l'attribut	Description
1	Nb_pkts	Nombre de paquets dans l'échantillon ΔT_i
2	Diff_ip_src	Nombre de différentes adresses IP source
3	Diff_ip_dst	Nombre de différentes adresses IP destination
4	Diff_mac_src	Nombre de différentes adresses MAC source
5	Diff_mac_dst	Nombre de différentes adresses MAC destination
6	Fragment	Nombre de paquets fragmentés
7	Diff_ports	Nombre de ports udp source et destination qui diffèrent
8	Pkts_max_sport	Nombre maximum de paquets envoyés par un seul port source
9	Pkts_max_dport	Nombre maximum de paquets reçus par un seul port destination
10	Port_max_ipsrc	Nombre maximum de ports source utilisés parmi tous les hôtes
11	Port_max_ipdst	Nombre maximum de ports destination utilisés parmi tous les hôtes
12	Taille_min	Taille minimum des paquets dans un échantillon (octets)
13	Taille_max	Taille maximum des paquets dans un échantillon (octets)
14	Taille_moy	Taille moyenne des paquets dans un échantillon (octets)
15	Taille_écart_type	Écart-type de la taille des paquets dans un échantillon (octets)
16	Nb_arp	Nombre de requêtes ARP non répondues
17	Nb_icmp	Nombre de paquets ICMP
18	Nb_tftp	Nombre de paquets TFTP
19	Nb_snmp	Nombre de paquets SNMP

Table 9 – Liste des attributs statistiques construits sur une fenêtre d'observation de durée ΔT

5.5. APPRENTISSAGE NON SUPERVISE POUR L'ÉTAPE 3 : *CLUSTERING* DE SOUS-ESPACE

5.5.1. *CLUSTERING* DE SOUS-ESPACE ET *LOCAL OUTLIER FACTOR*

Le *clustering* de sous-espace consiste à partitionner l'espace d'état en sous-espaces de dimension 2 ou 3 et à *clusteriser* dans chacun des sous-espaces. Ensuite les résultats obtenus suite au *clustering* de chacun des sous-espaces sont corrélés de sorte à ce que si un échantillon appartient au moins **N** fois à un *outlier* (cluster de taille 1) ou à un petit cluster d'une taille maximale **M**, cet échantillon sera considéré comme une anomalie. Il est évident que plus un cluster est grand et plus il est à même de contenir du trafic normal et donc potentiellement des faux positifs. Afin de réduire le nombre de faux positifs, en les distinguant de façon significative des vraies anomalies, nous avons choisi de calculer le Local Outlier Factor (LOF) pour chacun des échantillons labélisés comme anormaux. LOF est un algorithme de densité qui permet d'identifier les *outliers* locaux en comparant la densité locale d'un point à la densité locale de ses k plus proches voisins.

5.5.2. REDUCTION DES FAUX POSITIFS EN UTILISANT LE *LOCAL OUTLIER FACTOR* (LOF)

Nous avons testé plusieurs algorithmes de *clustering* : Affinity Propagation, DBSCAN, KMeans, MeanShift et Ward tirés de la librairie scikit-learn. La Table 10 donne les paramètres optimums obtenus par *grid-search* pour chacun de ces algorithmes, ainsi que les mesures de performance (colonnes 3 et 4) obtenus dans les captures contenant des anomalies. Nous avons évalué aussi la quantité de faux positifs trouvés dans les captures 100% normales, dans ce cas le F-mesure n'a pas de sens car il n'y a pas de vrais positifs, pour cela, nous considérons le Ratio de Faux Positifs (RFP), i.e. le ratio entre le nombre d'anomalies détectées et le nombre total d'échantillons (colonne 6). La table 11 présente exactement les mêmes tests mais cette fois en appliquant le LOF à chacune des anomalies. L'on constate que la quantité de faux positifs est significativement réduite en utilisant cette technique.

Algorithme de *Clustering*	Paramètres Optimums	Résultats utilisant des sets de données avec des anomalies		Résultats utilisant des sets de données sans anomalies	
		Précision	F-mesure	Précision	RFP
Affinity Propagation	dampling=0,9 conv_iterations=2 max_iterations=10	1,000	1,000	0,951	0,049
DBSCAN	eps=0,2 min_pts=7	0,960	0,842	0,928	0,072
KMeans	nb_clusters=7	0,983	0,922	0,996	0,004
MeanShift	Aucun	1,000	1,000	0,868	0,132
Ward	nb_clusters=6	0,98	0,95	0,934	0,066

Table 10 – Comparaison des algorithmes de clustering en termes de Précision et de F-mesure pour les sets de données contenant des anomalies ainsi que pour un set de données propre de toute anomalie avant utilisation du LOF

Algorithme de Clustering	Paramètres Optimums	Résultats utilisant des sets de données avec des anomalies		Résultats utilisant des sets de données sans anomalies	
		Précision	F-mesure	Précision	RFP
Affinity Propagation	dampling=0,9 conv_iterations=2 max_iterations=10	1,000	1,000	0,996	0,004
DBSCAN	eps=0,2 min_pts=7	0,967	0,864	0,996	0,004

KMeans	nb_clusters=7	0,983	0,922	0,996	0,004
MeanShift	None	1,000	1,000	0,996	0,004
Ward	nb_clusters=6	0,996	0,952	0,996	0,004

Table 11 – Comparaison des algorithmes de clustering en termes de Précision et de F-mesure pour les sets de données contenant des anomalies ainsi que pour un set de données propre de toute anomalie après utilisation du LOF pour réduire le nombre de faux positifs

Cependant, le principal problème à régler reste quand même l'efficacité de l'approche temps réel. En effet, la combinaison[11] de 2 parmi 19 attributs requiert 171 itérations. La figure 3 montre le temps requis pour le calcul du *clustering* se sous-espace et leur corrélation en fonction de la quantité d'échantillons à être traités. L'on observe que MeanShift n'est pas du tout adapté car il est bien trop lent.

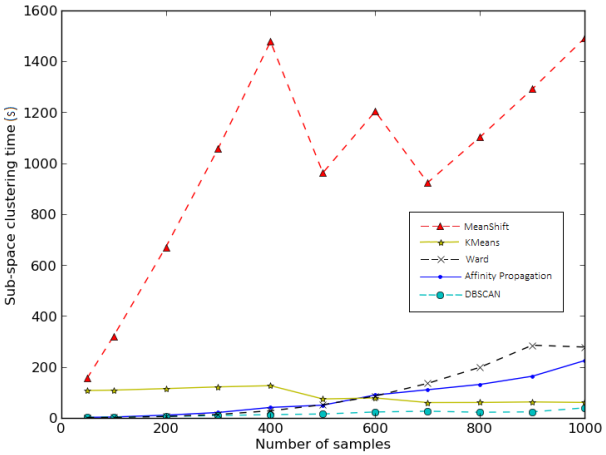


Figure 3 – Temps d'exécution du *clustering* de sous-espace et corrélation en fonction du nombre d'échantillons traités avec les algorithmes : MeanShift, KMeans, Ward, Affinity Propagation et DBSCAN

5.6. Proposition Supervisee pour l'Étape 3: One Class SVM

5.6.1. Parametres d'OCSVM

L'un des plus grands challenges dans la mise en application des Séparateur à Vaste Marge (SVM) est la calibration de ses nombreux paramètres. Les paramètres d'OCSVM se divisent en deux catégories : ceux spécifiques à la technique SVM et ceux relatifs au noyau :

Paramètres spécifiques à SVM :

- $\nu \in (0,1]$ représente à la fois le seuil supérieur du ratio d'*outliers* et le seuil minimum du ratio de vecteurs supports parmi tous les échantillons de la phase d'apprentissage.
- $\tau \in (0,1]$ ou tolérance représente le critère d'arrêt de la résolution asymptotique du problème quadratique.

[11] La combinaison de k parmi n se calcule comme suit : $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Paramètres du noyau. Le noyau linéaire ne requiert aucun paramètre. Mais puisqu'il est rarement applicable, il faut plutôt utiliser le noyau polynomial, RBF ou sigmoïdal qui ont les paramètres suivants :

- Le degré du noyau polynomial d ou bien le paramètre Gaussien $1/\sigma$ jouent un rôle important dans la flexibilité de la frontière de décision. Plus d ou $1/\sigma$ sont élevés et plus la frontière est courbée.
- Le coefficient γ du noyau polynomial donne l'angle de la courbure.
- La constante c du noyau polynomial permet d'ajuster l'offset.

5.6.2. CALIBRATION D'OCSVM

Pour déterminer le noyau et ses paramètres les plus adaptés au set de données, nous avons procédé par *grid-search* sur un set de test. Nous avons pris les paramètres suivant une échelle logarithmique (sauf pour le degré) entre 2^{-20} et 0,999 pour v et τ, entre 2^{-20} et 2^{20} pour gamma et le coef0, et entre 1 et 5 pour le degré. Il y a un autre paramètre à prendre en compte dans notre étude, c'est la fenêtre d'observation ΔT. Nous avons consigné tous les résultats du *grid-search* dans ces diagrammes en boîte pour chacun des noyaux. Chaque boîte représente les second et troisième quartiles[12] séparés par une médiane (rouge), et les lignes en pointillés représentent les premier et quatrième quartiles. Les croix en-dehors sont des *outliers*. Dans ce contexte, les résultats montrent qu'avec n'importe lequel des noyaux, l'on peut atteindre des performances acceptables. Cependant, le noyau RBF fournit les meilleures performances.

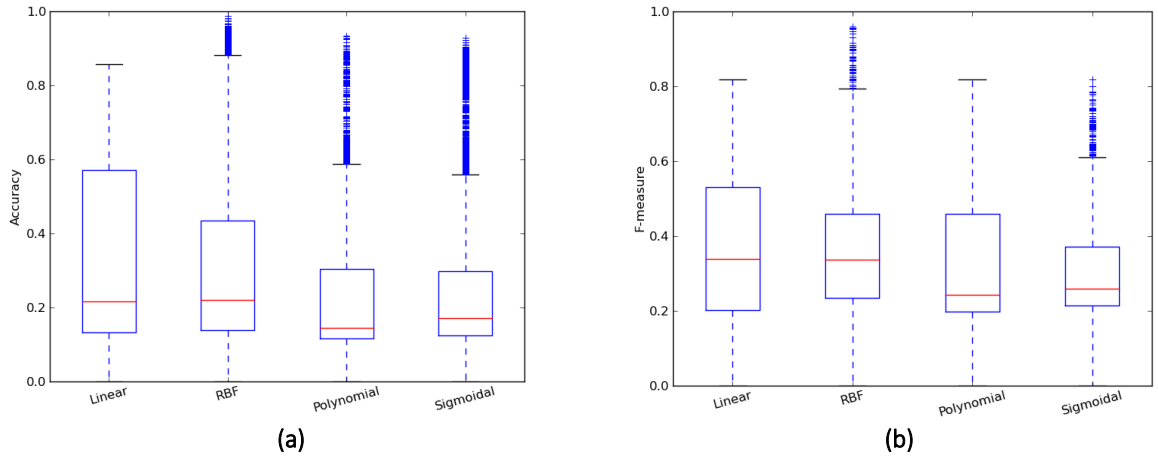


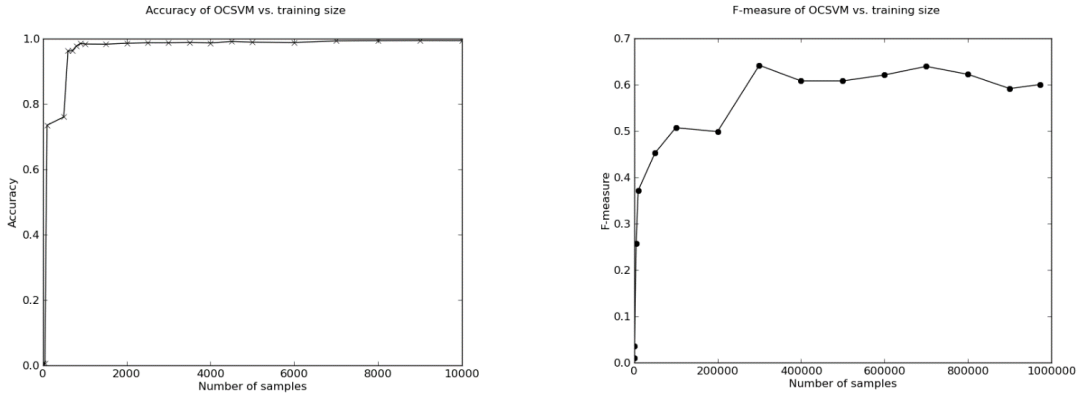
Figure 4 – Diagrammes en boîte de la Précision (a) et F-mesure (b) après le *grid-search* global pour chaque noyau

5.6.3. INFLUENCE DU VOLUME DE DONNEES D'APPRENTISSAGE SUR OCSVM

Etant donné qu'en l'absence d'anomalies, un nombre plus important d'échantillons normaux est requis, l'un des premiers aspects que nous voulions vérifier est le suivant : avons-nous assez de captures de trafic "normales" pour que l'algorithme One Class SVM puisse modéliser le trafic suffisamment fidèlement ? Pour ce faire, nous avons utilisé le set KDD'99, que nous divisé en deux : un set de test contenant des anomalies et un set 100% normal. On observe qu'avec un minimum de 600 échantillons,

[12] Quartile signifie que 25% des données se trouvent entre les deux valeurs représentées graphiquement par des boîtes.

la précision atteint un statut asymptotique (a). Le F-mesure n'a pas le même comportement, même avec 972.000 échantillons, l'on n'atteint pas d'asymptote.

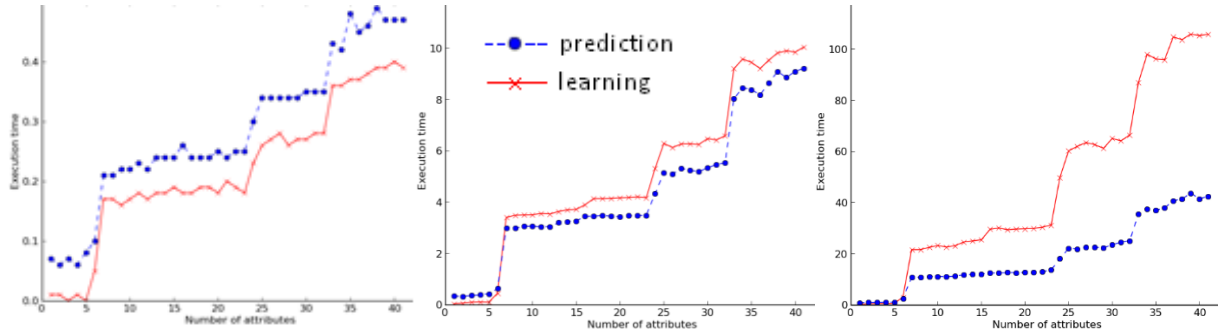


(a) (b)

Figure 5 – Influence de la taille du set d'apprentissage (en termes de nombre d'échantillons) sur la Précision de la détection d'anomalies (a) et le F-mesure (b)

5.6.4. TEMPS D'EXECUTION

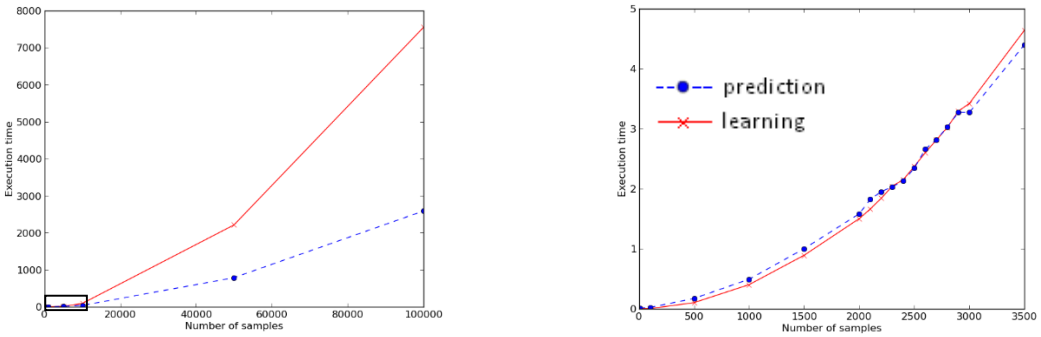
Dans cette partie, nous montrons l'influence du nombre d'attributs (fig. 6) et du nombre d'échantillons (fig. 7) sur le temps d'apprentissage d'OCSVM (en rouge) et de prédiction (en bleu). Pour l'influence du nombre d'attributs (fig. 6), nous avons pris arbitrairement 1000 (a), 5000 (b) et 10000 (c) échantillons.



(a) (b) (c)

Figure 6 – Temps d'apprentissage et de prédiction de l'algorithme OCSVM en fonction du nombre d'attributs

Curieusement, le temps de prédiction (points bleus) est légèrement supérieur que le temps d'apprentissage (a) quand le nombre d'échantillons est petit. Nous remarquons aussi que le temps d'exécution n'est pas proportionnel au nombre d'attributs, on peut clairement distinguer 4 étapes : entre 1 et 5 attributs, entre 7 et 25, entre 25 et 33 et entre 34 et 41. Le temps d'exécution dépend du nombre d'échantillons à traiter. Sur la figure 7, nous avons tracé l'influence du nombre d'échantillons en considérant la totalité des attributs pour 10 à 100.000 échantillons (a) et de 10 à 3.500 (b). Globalement, le temps d'apprentissage est inférieur au temps de prédiction (a), mais uniquement lorsque le nombre d'échantillons est au-dessus du seuil de 2.900 échantillons (b).



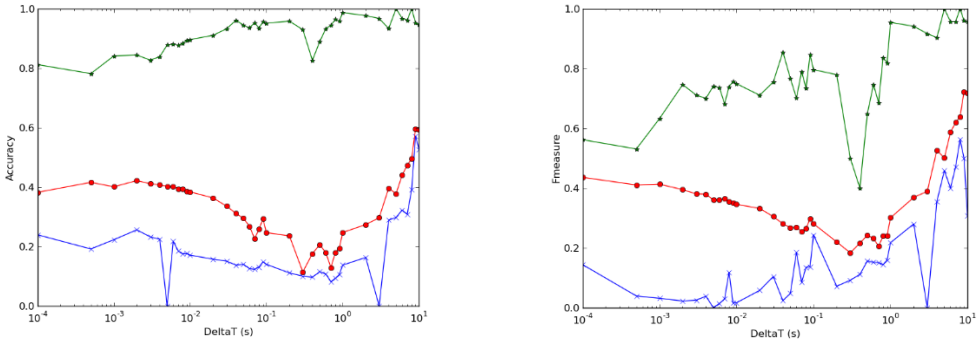
(a) (b)

Figure 7 – Temps d'apprentissage et de prédiction de l'algorithme OCSVM en fonction du nombre d'échantillons

5.7. INFLUENCE DE LA TAILLE DE LA FENETRE D'OBSERVATION

Les résultats figurant sur la figure 8 montrent clairement que la taille de la fenêtre d'observation ΔT a une très forte influence sur le F-mesure (b). Cependant, le noyau RBF a de bons résultats pour ΔT=1s. Nous avons obtenu une précision de 0,98 et un F-mesure de 0,95 pour γ=0,25 et ΔT=1s, et ce pour différentes combinaisons de ν et τ, plus concrètement avec 21 vrais positifs, 2 faux positifs et aucun faux négatif ce qui est un bon résultat, sachant que ces 2 faux positifs peuvent être éliminés avec le LOF. Nous avons fait de même pour la technique de *clustering* de sous-espace, pour lequel les performances sont optimales pour M=10 and N=20 et ΔT=1s (fig. 9).

Noyau RBF



(a) (b)

Figure 8 – Influence de la fenêtre d'observation pour la précision minimum (bleu), moyenne (rouge) et maximum (vert) (a-c-e-g) et F-mesure (b-d-f-h) de l'algorithme OCSVM pour le noyau RBF lors du *grid-search* avec différentes combinaisons de paramètres en fonction de la durée de la fenêtre d'observation ΔT

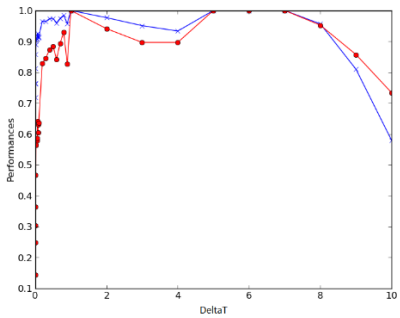


Figure 9 – Influence de la durée de la fenêtre d'observation sur la précision (croix bleues) et sur le F-mesure (points rouges) du *clustering* de sous-espace pour M=10 et N=20

CHAPITRE 6 : CONCLUSION

La sécurité pour la sûreté de vol sur les systèmes embarqués est un domaine émergent où beaucoup d'aspects restent encore à définir, et ce à différents niveaux. Il est nécessaire de protéger les avions existants contre les attaques, mais surtout il faut éviter d'introduire des vulnérabilités dans les futures architectures embarquées. Nous avons contribué dans ce domaine en proposant une méthodologie d'analyse de risque simple qui permet d'évaluer les risques de façon semi-quantitative et en accord avec les nouveaux standards de sécurité des systèmes embarqués en cours de construction. D'un point de vue plus technique, nous avons proposé une fonction d'audit du réseau bord pour assurer la maintenance en conditions de sécurité du réseau de l'avions. Le but de cette fonction d'audit, basée sur des techniques de Machine Learning (one class SVM, *clustering* de sous-espaces et Local Outlier Factor), est d'être générique et autonome afin de détecter des comportements anormaux dans le trafic réseau sans pour autant connaître la nature de l'attaque.

Dans ce chapitre, l'on résume chacune des contributions présentées dans cette thèse avec leurs respectives avantages et inconvénients. En conclusion, nous donnons des axes d'amélioration et les potentielles perspectives d'évolution.

6.1. CONTRIBUTIONS

6.1.1. METHODOLOGIE D'ANALYSE DE RISQUE

Dans cette dissertation, nous avons proposé une méthodologie d'analyse de risque simple et adaptable pour pallier au manque d'instructions pourvus par les standards en cours de construction (ED-202A, ED-203 et ED-204). Plus particulièrement, nous proposons un moyen d'évaluer le risque de façon semi-quantitative et déterminer l'acceptabilité des scenarios de menace à partir de leur impact sur la sûreté de vol et de leur vraisemblance, elle-même définie par la combinaison de la capacité de l'attaquant et l'exposition de l'*asset*, déterminés par des tables d'évaluation.

Avantages. Cette méthodologie présente l'avantage d'être systématique une fois tous les attributs et métriques définis. Elle est aussi très simple à appliquer, adaptable au contexte de l'étude et respecte les standards. Elle a été utilisée pour l'évaluation dans des cas réels et soumis aux autorités de certification en réponse à des requêtes d'évaluation du risque et a été approuvé comme une preuve valide d'analyse de risque préliminaire. Aussi, des ingénieurs de sûreté de fonctionnement du COMAC (*Commercial Aircraft Corporation of China*) nous ont contactés pour utiliser cette méthodologie, ainsi que Siemens AG à Braunschweig (Allemagne) a transposé cette méthodologie pour l'analyse de risque des systèmes ferroviaires.

Faiblesses et difficultés. Le niveau de subjectivité de cette méthodologie dépend encore de la taxonomie des attributs. Évidemment, l'analyse de risque en elle-même n'est pas suffisante et requiert d'être complétée par des tests d'intrusion pour valider ou infirmer les scénarios de menace identifiés. De plus,

il est objectivement compliqué d'évaluer une méthodologie puisqu'en théorie elle aurait dû être revue et testée par tous ses utilisateurs potentiels (experts en sécurité, processus, qualité, certification, architectes système, etc.) et ce pendant une longue période, ce qui n'était évidemment pas possible dans le contexte de cette thèse. La tâche était d'autant plus compliquée que les standards ED-202, ED-203 et ED-204 changent en permanence et nous ignorons si les concepts énoncés dans les premières versions seront encore valables dans les versions définitives.

6.1.2. Fonction d'Audit pour les Reseaux Bord

L'un des piliers pour assurer que les conditions de sécurité sont maintenues même en cas de cyber-attaque, c'est l'observation du réseau. Notre seconde contribution consiste en une fonction autonome de détection d'anomalies dans le réseau composée de quatre étapes : (1) capture des paquets, (2) conditionnement des paquets, groupement en échantillons et construction d'attributs, (3) classification des échantillons avec un algorithme d'apprentissage et (4) post-traitement. Pour les étapes 2 et 3, nous proposons différents blocs : deux façons de modéliser le trafic au travers d'attributs pour l'étape de conditionnement et l'utilisation soit d'une méthode d'apprentissage supervisée (algorithme One Class SVM) ou non-supervisée (*clustering* de sous-espace) pour modéliser le comportement normal du réseau et être capable de prédire si un échantillon est normal ou non.

Avantages. Nous avons travaillé avec des exemples d'attaques telles que des scans de réseau et des dénis de service qui sont très facilement détectables avec notre première proposition d'attributs, la deuxième liste d'attributs visait à détecter des attaques de rejeu de paquets avec un contenu modifié est encore à parfaire. Etant donné que le plus grand challenge dans la détection d'anomalies est de réduire le nombre de faux positifs générés par le système, nous avons proposé d'utiliser un coefficient, le Local Outlier Factor pour distinguer ces fausses alertes des vraies anomalies, ce qui a donné de très bons résultats.

Faiblesses et difficultés. Les résultats obtenus ont été satisfaisants compte tenu de la quantité insuffisante de données que nous avions pour les tests. En pratique, nous n'avions pas assez de trafic pour assurer que nos algorithmes apprenaient un modèle suffisamment fidèle. Avec un set de données plus important et plus exhaustif, les résultats seraient sans doute bien meilleurs.

6.2. Perspectives

Le premier aspect pour améliorer les performances de la fonction d'audit serait de modéliser le profil de comportement normal du réseau pour chacune des phases de vol pour plus d'exactitude, étant donné que le trafic peut varier selon si l'avion est au sol, en phase de décollage, atterrissage ou en croisière.

Un autre aspect serait d'utiliser une architecture redondante et dissimilaire souvent utilisée en sûreté de fonctionnement. Redondance pour augmenter la fiabilité du système en dupliquant et en parallélisant ses composants. Dissimilarité pour assurer l'intégrité du système en utilisant deux composants faisant la même fonction mais de fabrication différente. Concrètement, le système d'audit pourrait être redondé de part et d'autre des passerelles entre les différents réseaux de l'avion (ACD,

AISD and PIESD). La dissimilarité viendrait d'une part de l'utilisation de différents sets d'attributs et d'autre part, de différents algorithmes d'apprentissage. Les résultats de chaque système seraient ensuite corrélés pour arriver à la prédiction définitive.

Enfin, on pourrait coupler les alertes du système d'audit avec celles du Flight Warning System, c'est-à-dire le système de contrôle des pannes systèmes. L'objectif serait d'une part d'éviter les faux positifs en regardant de plus près s'il s'agit d'événements rares mais légitimes comme une alerte du système de contrôle de la sûreté du vol mais aussi de déterminer si une attaque a provoqué des pannes au niveau des systèmes avion.

Pour conclure, cette dissertation est juste un grain de sable dans l'immense quantité d'activités à mettre en place pour assurer que l'avion restera dans un état sûr même en cas de cyber-attaque. Si les architectures des systèmes embarqués continuent à évoluer de la sorte, cela donnera du fil à retordre aussi bien aux hackers, qu'aux futurs thésards qui vont s'attaquer au sujet !