Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

L'analyse de risque impactée par l'apprentissage machine
Impact of Machine Learning on risk analysis

Frédéric Deschamps
Direction Technique et Innovation
LGM
Toulouse, France
frederic.deschamps@lgm.fr

Nicolas Rémy
Ingénierie et Performance des Systèmes
LGM
Vélizy, France
nicolas.remy@lgm.fr

Arnault Ioualalen
CEO
NUMALIS
Montpellier, France
ioualalen@numalis.com

***Résumé* : l'intégration d'apprentissage machine dans des applications critiques remet en cause les pratiques actuelles de démonstration de niveau de sécurité. L'approche répondant aux attentes d'un niveau de sécurité acceptable n'existe pas à ce jour. L'acculturation de la communauté de maîtrise des risques est nécessaire pour être en mesure de juger de l'acceptabilité des arguments. L'illustration des concepts traitée au travers d'un cas d'usage simple permet aux différentes communautés d'échanger autour d'une même référence.**

Summary: the integration of machine learning into critical applications challenges current safety level demonstration practices. An approach covering the expectations of a security analysis does not exist to date. The acculturation of the safety community is necessary to be able to judge the acceptability of the safety arguments. An illustration of the concepts treated is proposed through a simple use case. It allows the different communities to exchange around the same reference.

Mots-Clés : Risque, Sécurité, Intelligence Artificielle.

I. OBJECTIF

Cette communication propose de rappeler les attendus d'une étude de sécurité et de les mettre au regard des propriétés d'algorithmes basés sur l'apprentissage machine (AM) équivalent au concept de « Machine Learning » (LM). Des approches de démonstration sont alors proposées et discutées. Le point de vue proposé est celui de l'analyste en Maîtrise des Risques (MdR).

En complément, une application est proposée afin d'illustrer les algorithmes d'intelligence artificielle et leur mise en œuvre.

II. CONTEXTE ET PROBLEMATIQUE

Dans de nombreux secteurs, une méthodologie de maîtrise des risques reconnue est documentée (CS25 et ARP pour l'aéronautique, ISO26262 pour l'électronique automobile, les normes NF EN 50126/8/9 pour le ferroviaire, etc. …). Parfois, l'activité donne lieu à une certification qui implique un tier, délégué d'un état, prononçant une conformité à un ensemble de règles[1]. Ce contexte normatif et réglementaire implique que les preuves soient compréhensibles par ces intervenants. Les principes sous-jacents aux méthodologies d'analyse de sécurité ne sont pas tous explicites, et peuvent être violés par les spécificités de l'AM.

Les performances de certains algorithmes d'AM comme le Deep Learning (DL) sont telles qu'il est inévitable de les implémenter dans des applications critiques. Il est donc nécessaire de construire un argumentaire permettant de justifier du niveau de sécurité d'un système intégrant de l'AM. Cependant, sans se laisser aveugler par les performances étonnantes de l'AM, il n'est pas établi que cette démonstration soit possible sans modification profonde des méthodologies de démonstration de sécurité, ou sans acceptation de risques résiduels.

Au même titre que de nombreuses sociétés, LGM est confrontée de plus en plus souvent à l'usage de l'AM dans des applications, et doit souvent se prononcer sur l'acceptabilité de la solution.

Cette analyse est le résultat de cette expérience, complétée de la revue d'une partie de l'abondante littérature sur le sujet. Notre contribution comporte 5 étapes :

1. Typologies des applications rencontrées par LGM.
2. Rappels des objectifs, et méthodes d'une analyse de sécurité.
3. Rappel des caractéristiques de l'AM impactant la démonstration de sécurité.
4. Liste des défis posés par l'AM à la démonstration de sécurité.
5. Proposition d'une classification simple des approches possibles pour répondre à une

[1] Une définition plus développée et précise de la certification est proposée dans [3].

demande de qualification d'une application utilisant de l'AM.

En complément, LGM partage un cas d'usage utilisé en interne pour montrer la puissance de l'AM et les défis posés à une analyse de sécurité. Cet exemple est apparu nécessaire pour aider les échanges. Il permet de discuter autour d'un cas réel, et d'essayer les recommandations ou moyens de mitigations. LGM l'a voulu « opensource » [1] afin de permettre à chacun de l'adapter à ses besoins.

III. TYPOLOGIE D'APPLICATIONS

Les applications de l'AM se développent rapidement, et LGM se trouve confrontée à cet usage lors des activités de démonstration de sécurité. Nos équipes l'utilisent aussi pour améliorer l'offre de service, notamment pour la maintenance préventive. L'expérience acquise à partir de ces activités est importante pour définir une démarche pragmatique.

En 2016, LGM a été sollicitée par un constructeur afin de qualifier un logiciel basé sur un réseau de neurones d'apprentissage profond permettant à un véhicule, de traverser une route, dans le cadre de l'ISO26262. En l'absence de méthodologie établie, il n'était pas possible de proposer une réelle démonstration de sécurité, seules quelques communications traitaient ces sujets, mais sans définir de méthode. En 2017, LGM a développé un algorithme basé sur des réseaux de neurones Long Short-Term Memory (LSTM) afin d'identifier des typologies de routes à partir d'accéléromètres. L'évaluation de la fiabilité des résultats a trouvé un début de réponse avec la matrice de confusion [7]. En 2019, l'utilisation d'IOT [2] afin d'instrumenter d'anciens véhicules militaires (VAB [3]) a permis de construire sur plusieurs mois une base de données de mesures. Elle vise à construire une estimation de la RUL (« Rest of Useful Life ») pour orienter la maintenance préventive. Fin 2019, l'utilisation de réseaux de neurones convolutifs (en apprentissage supervisé) a permis de créer des cartes pour une application ferroviaire ; dans ce cadre LGM intervient pour réaliser une étude de sûreté de fonctionnement. La même année une étude sur les fiches de faits techniques à partir de plusieurs algorithmes a été menée afin de justifier de l'extension du délai entre deux visites de maintenance préventive. En 2020, LGM a finalisé l'étude pilotée par l'IMDR, visant à analyser l'impact des « Big Data » sur la sûreté de fonctionnement [19].

TABLE I. SYNTHÈSE DES CAS RENCONTRÉS

Secteur	***Objectif***	***Approche retenue***
Automobile	Cas 1 - Trajectoire autonome (avec fusion données LIDAR[4], GNSS[5] et possibilité d'ajouter des prises de vue) dans cadre ISO26262. Impact sécuritaire. Besoin d'analyse d'image pour reconnaissance des panneaux et de la configuration de la route.	Aucune. Pas de méthode disponible.
Automobile	Cas 2 - Algorithme pour identifier le type de route pour des véhicules. Pas d'aspect sécuritaire.	Analyse matrice de confusion (Réseau de type LSTM).
Militaire	Cas 3 - Analyse vibration, courant, température, vitesse, position pour approche de type Health Monitoring. Pas d'aspect sécuritaire.	Diverses méthodes (ML et « standards »). % de bonne classification.
Ferroviaire	Cas 4 - Création d'une carte ferroviaire à partir de mesures LIDAR dans cadre EN 50128. Impact sécuritaire.	Taux de vrais positifs et taux de faux positifs. Monitoring de sécurité.
Aéronautique	Cas 5 - Analyse qualité d'analyses de rapports de maintenance préventive (fiches de faits techniques) dans le cadre d'IP44 visant à allonger le délai entre deux visites avion. Impact sécuritaire.	Test de plusieurs algorithmes avec le logiciel Dataiku.
Multi-secteurs	Cas 6 - Analyse l'impact des « Big Data » sur les pratiques en sûreté de fonctionnement. Cette analyse a mené à un cas test.	Divers algorithmes d'AM pour analyser des rapports de maintenance.

Nous proposons une taxonomie simple des cas rencontrés :

A. Cas d'utilisation sans impact sécuritaire (cas 2, 3).

B. Cas d'utilisation avec un impact sécuritaire.

Ce cas se compose en deux catégories selon le délai entre une sortie erronée de l'algorithme d'AM, et un effet négatif sur la sécurité.

i. Cas B(i) - Un opérateur a le temps de prendre connaissance de la sortie de l'algorithme et d'agir avant qu'un effet négatif se produise (cas 5, 6[6])

ii. Cas B(ii) - L'opérateur n'est pas dans la boucle de décision (cas 1, 4[7]).

Le traitement du cas B(i) implique une situation de codécision entre un opérateur et l'algorithme qui permet de limiter l'impact d'une erreur. Ce cas est abordé dans [13] dans la cadre de la classification supervisée et n'est pas développé dans cet article. L'analyse porte sur le cas B(ii). Le cas d'usage - contrôle d'un véhicule autonome - a été choisi car il regroupe un grand nombre de difficultés rencontrées lors de nos études (régression, modèle boîte noire, difficulté de construction d'une matrice de confusion).

IV. PROPRIETES D'UN SYSTEMES SUR

Selon l'ISO, la sécurité se définit comme « *l'absence de risque inacceptable* » [14]. Mais le « risque 0 » n'existant pas, il est souvent question de risque tolérable, à savoir *« le risque accepté dans un certain contexte et fondé sur les valeurs admises par la société »* [14].

Le risque est « *la combinaison de la probabilité d'un dommage et de sa gravité* » [14]. La matrice de risque Fig. 1 est l'outil privilégié pour juger de l'acceptabilité d'un risque.

[2] Internet Of Objects

[3] Véhicule de l'Avant Blindé

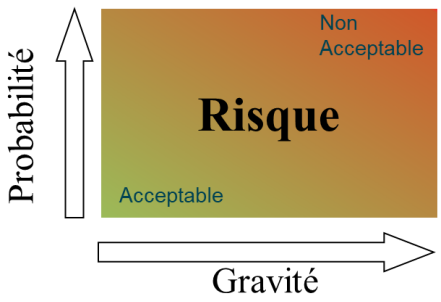
[4] LIght Detection And Ranging

[5] Géolocalisation et Navigation par un Système de Satellites

[6] Pour le cas test étudié. L'analyse de l'état de l'art est plus étendue.

[7] Une fois la carte embarquée dans un train autonome, l'opérateur n'est plus dans la boucle de décision.

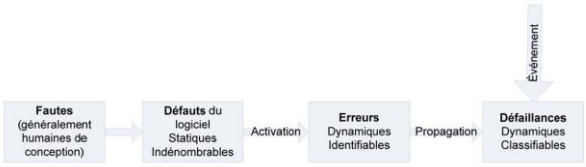
Fig. 1. Matrice de risque



La probabilité peut être estimée pour les pannes aléatoires, il s'agit de probabilité physique objective [35]. Elle n'est pas liée au degré de connaissance mais à la nature de l'objet étudié. Il s'agit, par exemple, d'une défaillance d'un composant électronique : quelques soient les choix de définition, le composant peut avoir une défaillance.

En revanche la probabilité ne peut pas être estimée pour un logiciel car la faute présente dans le logiciel se déclare si un évènement déclencheur se produit. L'humain créée une erreur qui s'active dans certain contexte pour mener à une défaillance système. Il est questions dans ce cas de pannes systématiques qui donnent lieu à une démonstration déterministe et qualitative [38].

Fig. 2. Principe de faute, défaut, erreur et défaillance



À ce jour de nombreuses normes traitent de la maîtrise des pannes systématiques au travers d'activités de contrôle et modélisation. La table suivante donne un extrait. Il est possible de la compléter par de nombreuses normes et règlements pour ces domaines ou d'autres (nucléaire, normes machines, militaire, naval, …).

TABLE II. LISTE DE NORMES DE SECURITE EN LIEN AVEC LES CAS D'USAGE RENCONTRES [8].

Domaines	**Normes**	
	Identifiant	***Couverture***
Aéronautique	ARP4754A	Avion (système de systèmes) Systèmes Equipement [a] Pannes systématiques
	DO178C	Logiciel Pannes systématiques
	DO254	Matériel (FPGA ou carte électronique) [b] Pannes systématiques
	ARP4761	Avion Système Equipment Pannes aléatoires Norme orientée méthodologie (AMDEC, arbres de défaillances, analyse de pannes communes, …)
Automobile	ISO26262 ver. 2018	Système carte électronique, composant complexe, logiciel
	SOTIF	Safety Of The Intended Functionality Lorsqu'une fonction ne couvre que partiellement le comportement attendu.
Ferroviaire	510126	Approche système
	50128	Logiciel
	50129	Construction du dossier de justification
Militaire	MIL-STD 882E	Approche système Allocation de niveau pour le logiciel

[a] Parfois la norme ARP4754A est utilisée comme référence pour le développement d'un équipement électronique à la place de la DO253

[b] L'applicabilité est systématique aux composants complexes comme les FPGA. L'applicabilité aux cartes électroniques donne lieu à des notes d'application issues des différentes autorités et peuvent varier selon les pays.

Certains articles, comme [27], concluent que l'AM est hors périmètre de l'ISO 26262. La norme SOTIF, développée dans le cadre de l'automobile, propose quelques éléments de réponses, mais ne couvre pas, à notre connaissance, tous les aspects du sujet. Il est donc nécessaire de définir une méthodologie pour adresser la problématique posée par l'AM.

La robustesse du système vis-à-vis des pannes systématiques repose sur trois principes[9] qui sont combinés dans les normes précédemment citées (comme l'ARP4754A) :

1. évitement des fautes afin d'empêcher l'occurrence de défaillances, en se basant, par exemple, sur des règles de codages ;
2. élimination des fautes en détectant, identifiant et enlevant les fautes du système (fautes et défauts), en se basant, par exemple, sur des tests ;
3. tolérance aux fautes afin de préserver le service malgré l'occurrence d'erreurs, en se basant, par exemple, sur une architecture système limitant l'effet d'une faute.

Afin de répondre à ces trois activités, l'algorithme, pour être analysé, doit remplir, certaines caractéristiques [18] [28] [10]:

- Être transparent. Cette caractéristique renvoie « *à notre capacité à comprendre les décisions prises par un algorithme, à nous assurer qu'il fasse bien ce qu'il est censé faire, et à éviter tout effet inéquitable* » [28]
- Être explicable. « *Un algorithme est explicable, s'il est possible de donner à l'ensemble des utilisateurs, quel que soit leur bagage éducatif [11], une vision claire des procédures employées et des fonctionnalités remplies par l'algorithme, afin de permettre un usage informé* » [28].
- Être intelligible. « *Un algorithme est intelligible s'il est possible au concepteur de comprendre*

[8] Une analyse comparative multi-secteurs, dans le cadre classique, est proposée dans [36], [37], [38].

[9] D'après [38], il est question de « *correctness* » qui revient à l'évitement et l'élimination des fautes : « *the avoidance of software failures (thanks to the avoidance or elimination of their causes, the errors or faults in software* »

[10] Il existe d'autres caractéristiques d'un algorithme comme la loyauté ou l'équité. Mais elles ne nous semblent par pertinentes dans notre contexte.

[11] Cette définition, générale, est à moduler. Dans le cadre d'une étude de sécurité, il est possible de demander un bagage minimum, mais pas spécialisé dans l'AM.

son fonctionnement et de vérifier s'il satisfait bien les propriétés désirées » [28].

Ces trois propriétés sont nécessaires pour mener, à notre avis, une étude de sécurité. Elles sont difficiles à décliner en termes de règles de décision, mais elles sont un guide pour définir une démarche. Ces propriétés peuvent être applicables au niveau local (sur une seule prédiction) ou au niveau global (sur tout le fonctionnement de l'algorithme). Le niveau local est très certainement atteignable (approche formelle, « heatmap »), mais l'approche générale est celle qui offre la plus grande difficulté. Les limites des réseaux vis-à-vis de ces concepts sont analysées dans les sections suivantes.

Le but d'une analyse de sécurité est de convaincre des individus qu'un système est sûr. Il nous semble utile d'explorer rapidement ce point de vue. Les études de sécurité ont deux objectifs [2] :

1. d'une part elles consistent à s'assurer que la définition d'un système est sûre,
2. et d'autre part elles consistent à convaincre les autorités qu'un niveau de sécurité acceptable est atteint.

La distinction est importante car le premier objectif est une caractéristique qui, finalement, se révèle être atteinte ou non lors de l'utilisation sur la vie du produit. Le travail des équipes consiste à s'assurer que la sécurité est bien prise en compte lors du développement (ingénieur conception, qualité, test, sécurité …). L'analyste de sécurité a un rôle important, mais il n'intervient souvent qu'en sanction d'une proposition de définition.

Le second objectif nous intéresse particulièrement, car il consiste à transmettre la vision des intervenants à des décideurs vis-à-vis l'acceptabilité du niveau de sécurité du produit. C'est ce second point sur lequel porte la présente analyse : il s'agit de se convaincre et de convaincre les parties prenantes que le système est sûr.

V. CARACTERISTIQUES DE L'APPRENTISSAGE MACHINE

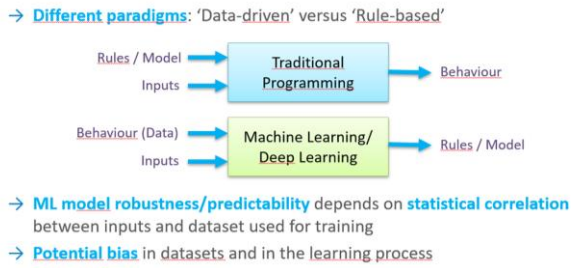
L'apprentissage machine est une nouvelle façon de programmer des applications. L'EASA propose des définitions et une taxonomie intéressante dans son rapport [10]. Cette section s'appuie largement sur ce document.

Les applications d'Intelligence Artificielle (IA) peuvent être divisées entre :

- IA symbolique - L'IA est basée sur les modèles (« model-driven AI »). Il s'agit par exemple des systèmes experts qui sont programmés de façon traditionnelle et s'appuient sur la logique pour déduire des conclusions d'assertions initiales.
- IA statistique – L'IA est basée sur les données (« data-driven AI »).

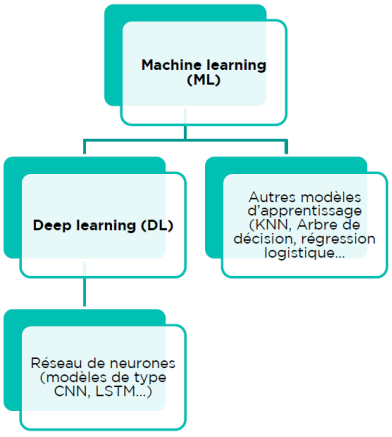
La percée actuelle porte sur l'IA statistique avec le « Machine Learning », aussi nommé apprentissage machine (AM). Dans ce paradigme de programmation, ce sont les données qui permettent de créer des règles (pas nécessairement intelligibles) de l'algorithme.

Fig. 3. Principe de l'apprentissage machine (ou machine learning) [10]



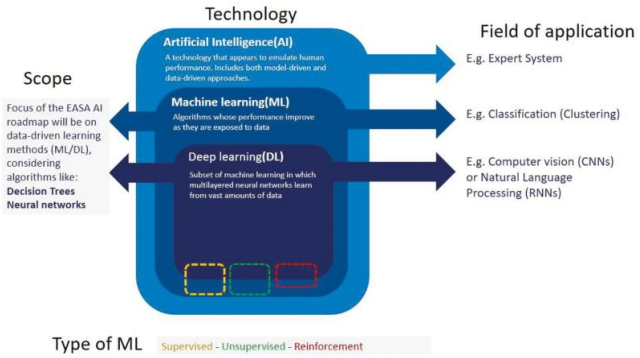
Les algorithmes se répartissent en différentes catégories.

Fig. 4. Catégories d'algorithmes



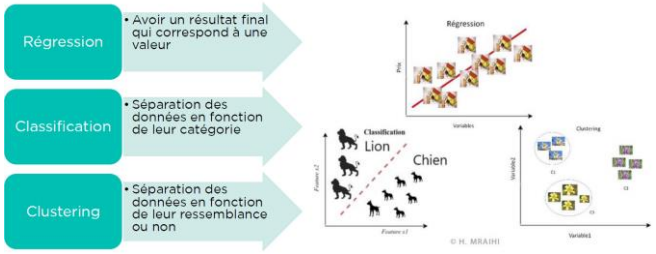
L'apprentissage profond (ou « Deep Learning ») (DL) permet d'accéder à des capacités approchant celles de l'humain dans certains domaines avec, par exemple, la reconnaissance d'images ou l'analyse automatique du langage.

Fig. 5. Taxonomie de l'IA [10]



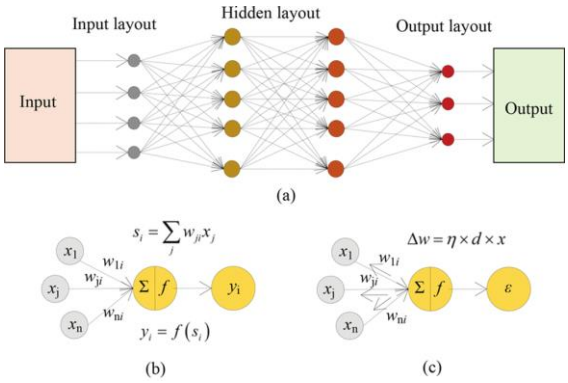
Les algorithmes peuvent être utilisés de différentes façons en fonction du type d'apprentissage sélectionné.

Fig. 6. Types d'apprentissages



Une attention particulière est portée sur les réseaux de neurones profonds car ils ont des caractéristiques particulières et de très bonnes performances. La principale caractéristique est que le programme produit est proche de celui de la boîte noire[12], donc les techniques habituelles de développement logiciel pour les applications critiques ne s'appliquent pas.

Fig. 7. Réseau de neurones profond[13]



(a)Exemple de réseau de neurones avec 2 couches cachées
(b)Les poids (w) permettent de pondérer l'influence de chaque sortie des neurones en entrée (gris) du neurone représenté (jaune).
(c)Principe de rétropropagation pour apprentissage du réseau.

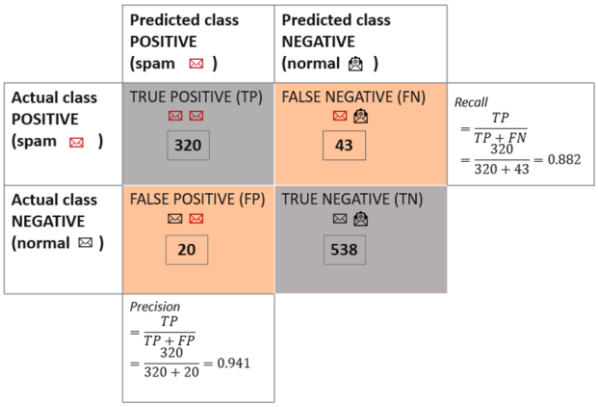
La phase d'apprentissage implique une phase d'optimisation qui minimise ou maximise une fonction. Cette fonction s'appelle la fonction objective (« objective fonction » ou « criterion »). Lorsque cette fonction est minimisée, il est question de fonction de coût (« cost function », « loss function » ou « error function »). Le principe consiste à minimiser la fonction en modifiant les poids dans le réseau par petits pas. Cette approche se nomme la descente de gradient. Souvent l'analyste obtient une courbe représentant l'évolution de la courbe d'erreur en fonction du nombre d'itérations. C'est un premier indicateur de la qualité de l'apprentissage.

Deux erreurs liées à la capacité de généralisation d'un réseau de neurones sont à éviter lors de l'apprentissage :

- Sous-apprentissage – Dans ce cas l'erreur est trop élevée donc le réseau généralise mal et ne sort pas les résultats attendus.
- Sur-apprentissage (« overfitting »). Dans ce cas le réseau colle parfaitement aux données d'apprentissages, mais ne généralise pas.

La matrice de confusion est un outil privilégié pour appréhender les performances dans le cas de classifications.

Fig. 8. Exemple de matrice de confusion[14]



Cette matrice est difficile à construire dans le cas d'une régression. En prenant l'exemple d'une consigne d'angle volant générée par un réseau en fonction des caméras pour une voiture, il est possible pour une même image de proposer plusieurs angles. Tant que l'angle permet de rester sur la route, le résultat peut être considéré comme juste. Dans ce cas la construction de la matrice est extrêmement difficile.

VI. LES LIMITES IMPOSEES PAR LA NATURE DE L'AM

L'apprentissage machine peut impacter la phase d'identification du risque. En effet, il n'est pas exclu que l'AM apporte de nouveaux risques [5][13]. Ce point ne semble pas bloquant et des solutions sont proposées [13].

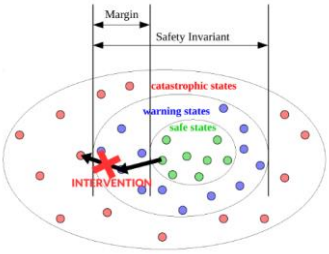
La phase d'allocation est aussi impactée selon que l'allocation sur l'apprentissage machine soit portée par le système ou le logiciel.

La question porte principalement sur la preuve de sécurité. Le cas d'usage montre que l'étude peut porter sur une preuve s'appuyant sur un ensemble de règles comme pour le logiciel, ou sur une démonstration de type probabilité comme pour les pannes aléatoires.

La proposition 3 de tolérance aux fautes est celle préconisée par [15] avec l'implémentation d'un moniteur. Mais la construction de ce dernier n'est pas toujours possible, car il faut qu'il soit capable, avec un algorithme déterministe[15], de maintenir le système dans un état sûr avec par exemple une architecture COM-MON (pour Commande et Monitoring), Fig. 9.

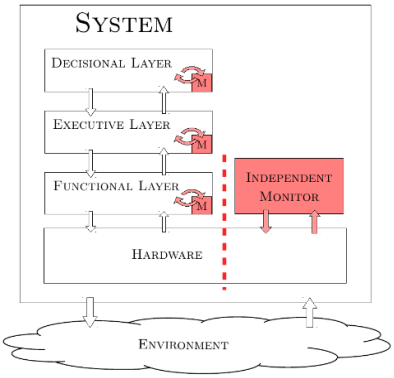
[12] Un réseau de neurones comme alexnet peut compter 60 millions de paramètres. [33].
[13] Schémas repris d'un site dont le lien a été perdu.
[14] Schémas repris d'un site dont le lien a été perdu.
[15] Il est nécessaire de montrer que l'algorithme de monitoring couvre tous les cas. Ces exigences ne nous semblent pas atteignables à ce jour.

Fig. 9. Principe des état de sûreté [15]



Il n'est souvent pas possible de définir un algorithme déterministe pour permettre de rester dans un domaine de sécurité. C'est pourquoi nous pensons que cette solution, si elle doit être privilégiée, n'est pas applicable à tous les cas rencontrés. L'implémentation est présentée dans Fig. 10.

Fig. 10. Monitoring implementation [15]

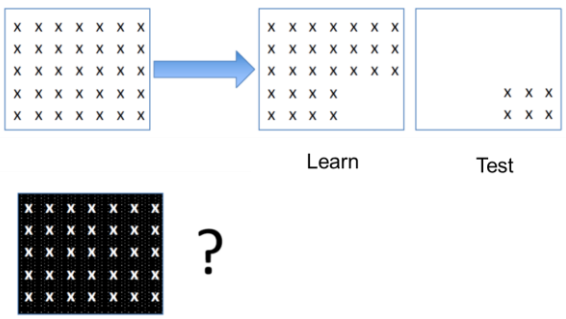


A. *Limites de l'apprentissage machine*

En revanche, il n'existe pas, à notre connaissance, de moyen de répondre aux points 1 et 2 (évitement et élimination des fautes) de façon exhaustive :

- L'apprentissage se base sur un jeu de cas qui, par essence, ne peut couvrir tous les cas possibles [2],[4],[6]. Il est possible de juger de la pertinence des cas par revue, de l'enrichir par simulation, voire de vérifier en temps réel si les cas proposés ne sont pas significativement différents des cas tests ; mais ces méthodes ne permettent pas de construire une confiance suffisante.

Fig. 11. Problème de représentativité des set d'apprentissage et de test[16]



- À ce jour la communauté ne comprend pas comment certains algorithmes, comme les réseaux de neurones convolutifs, généralisent la connaissance [16]. Cette interrogation implique une confiance limitée dans l'algorithme et dans sa capacité à suivre un comportement attendu. Il pose la question du manque de robustesse.
- La vérification d'exigence devient difficile [6]. En effet comment vérifier par exemple que l'algorithme peut détecter un passant et le distinguer d'un tas de feuilles, afin de ne pas arrêter le véhicule l'automne lorsque le LIDAR détecte des obstacles ? Il n'est pas possible de caractériser totalement un passant. À titre d'exemple, l'accident impliquant un véhicule UBER renversant une passante poussant son vélo. Le programme avait été « réglé » pour éviter les faux positifs pour assurer une continuité de service, mais cela s'est fait au détriment de la sécurité.
- Éviter le contournement du critère à optimiser ou « avoid reward hacking » [6]. L'algorithme va dans ce cas choisir un élément qui n'est pas représentatif. Il s'agit par exemple du cas où un AM confond le loup et le chien car il n'identifie les loups que par rapport à la neige en fond d'image.

Fig. 12. Problème de représentativité des set d'apprentissage et de test[17]

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

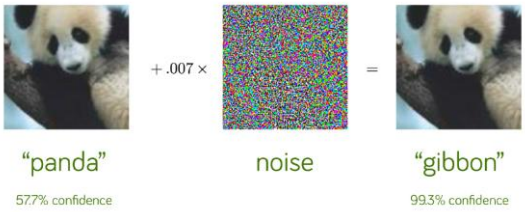
Table 2: "Husky vs Wolf" experiment results.

- La sensibilité au bruit, ou aux attaques des réseaux de neurones est un réel problème. La modification de quelques pixels bien choisis peut mener à changer la sortie avec un coefficient de confiance élevé, sans que cela ne soit détecté à l'œil nu. C'est une limite importante, même si des approches comme celle de Numalis ou ETH Zuric [29] permettent de mettre en avant dans les images des zones sensibles, dont une légère modification peut modifier significativement la valeur de sortie du réseau. Cette limite rend le réseau sensible au bruit, mais aussi aux attaques.

[16] Figure reprise de la présentation de Numalis ISCLP 2019 à Toulouse.

[17] Issu du site https://hackernoon.com/dogs-wolves-data-science-and-why-machines-must-learn-like-humans-do-41c43bc7f982 . Source primaire de l'image non trouvée.

Fig. 13. Problème de représentativité des set d'apprentissage et de test [26]



D'autres limites, qui semblent surmontables, sont identifiées comme le cycle de vie différent pour l'AM [5], comme les langages utilisés pour l'AM (tels que Python) qui ne sont pas couverts par les approches actuelles [5][4].

B. Contribution à la démarche de démonstration de la sécurité

Compte-tenu des limites exposées, l'approche se basant sur un argumentaire visant un respect strict des propriétés d'un système sûr semble voué à l'échec. Il est néanmoins possible d'avancer en construisant de nouveaux moyens acceptables de démonstration de sécurité.

La première recommandation consiste à ne pas intégrer les études des algorithmes de type apprentissage machine dans la démarche dédiée aux démonstrations logiciels. Elles doivent être intégrées dans la démarche système. L'EASA propose même de définir un niveau d'assurance particulier [10], sur le modèle des DAL (Développement Assurance Level) pour le logiciel. Mais les règles à établir ne sont pas définies et le problème reste ouvert. Le rapport de la Nasa envisage une approche qui rapprocherait l'AM de l'humain en mettant en place une licence dans l'esprit de celles accordées aux pilotes [6]. Là encore, nous doutons qu'une telle approche porte une confiance suffisante pour être proposée dans des fonctions critiques. Enfin, sans que nous ayons pu trouver de sources publiées, une approche consiste à considérer la performance de l'AM comme les pannes aléatoires du matériel électronique. Il serait dans ce cas possible d'estimer une probabilité d'occurrence et de renforcer les performances avec des architectures de type redondance ou « voteurs ». Une fois encore, étant donné la nature épistémique de l'incertitude, nous pensons qu'il s'agit d'une piste qui ne permet pas d'assurer un niveau de sécurité suffisant. Cependant, il est probablement plus facile de démontrer que des algorithmes ont des propriétés différentes et donc complémentaires (en utilisant des algorithmes dissimilaires), que de démontrer que l'algorithme final va « trouver » le concept de chat.

La caractérisation du comportement d'un système d'AM et plus particulièrement d'un réseau de neurones profond reste un problème difficile. Dans beaucoup de cas la définition même de leur domaine d'emploi est trop complexe car les concepts sous-jacents ne peuvent être formalisés. Il en résulte que le domaine d'emploi correspond à celui de données « similaires » à celles disponibles dans la base d'apprentissage. Sur ces données la notion de performance peut avoir un sens car le réseau est censé pouvoir généraliser correctement si sa phase d'apprentissage est réussie. La question des biais [34] que ces données peuvent avoir se pose, ainsi que celle des vulnérabilités que le réseau peut poser à des attaques à cause d'elles. Mais plus fondamentalement cela pose le problème de ne tester qu'à partir de ce que le concepteur connaît ou présuppose du système. Idéalement pour les applications les plus critiques il importe de garantir la sûreté du système pour toutes les données dans le domaine d'emploi. Cette approche est en général impossible à mettre en œuvre en pratique pour de l'apprentissage profond. Néanmoins des travaux récents [29], comme ceux de Numalis, montrent que des approches formelles permettent de démontrer des propriétés de sureté sur des parties de ce domaine. Ce type d'approche mathématique ne reposent pas sur du test mais sur la mécanique de preuves formelles par interprétation abstraite. Ce type de technique a déjà été utilisé en aéronautique pour démontrer la sûreté de code critique dans le passé. Ce n'est que récemment que ces techniques ont fait l'objet d'adaptation pour traiter le cas de l'apprentissage profond. Il est, par exemple, possible de démontrer qu'autour d'un point connu il n'existe pas d'attaque possible pour mettre en défaut la classification du système (stabilité du comportement). Il est également possible de démontrer les corrélations qui existent sur un domaine entre les entrées et les sorties afin d'expliciter une partie des règles internes qui restent inintelligibles en temps normal. Ces techniques innovantes font aujourd'hui l'objet d'évaluation et sont considérées au niveau normatif comme des pistes sérieuses pour garantir la robustesse de systèmes d'apprentissage profond (projets ISO 24029-1 et 24029-2 au sein de l'ISO/IEC JTC1/SC 42).

Les études sont nombreuses au sujet de la sécurité de l'AM au travers du monde. Des percées scientifiques peuvent invalider les conclusions si de réelles preuves mathématiques de stabilité se font jour. D'ici là, nous préconisons une acculturation de la communauté de la MdR pour porter un regard critique sur les propositions de démonstration de sécurité. Nous avons fait le choix d'appuyer cette présentation sur un cas test : construction d'un réseau de neurones visant à piloter un véhicule autonome.

VII. APPROCHES

A ce jour, les différentes approches comme l'IA explicable (DARPA), certifiable (DEEL) ne sont pas mûres. En l'état actuel, les approches possibles reposent sur des méthodes mises en œuvre pour des algorithmes déterministes :

- Si l'impact n'est pas sévère, il est possible au travers de différentes méthodes de gagner suffisamment de confiance pour utiliser l'AM. L'intérêt d'utiliser l'AM dans ce contexte est de plus d'acculturer le monde de la Maîtrise des Risques à ce sujet et de progresser dans la compréhension des limites. Ce sont, par exemple, les cas 2 et 3 de TABLE I.

Pour les autres cas :

- Ajout d'un moniteur de sécurité

  Cette approche est utilisée pour le cas d'usage n°4 (génération de carte).

  Si le monitoring couvre tous les cas de dysfonctionnents ou de comportements non attendus (i.e. permettant de rester dans un état de sécurité acceptable), alors l'AM peut être utilisée. Cela peut tout de même poser des problèmes : par exemple l'ARP4754A impose pour une fonction catastrophique que des

fonctions contributrices soient développées en DAL C minimum. En l'état l'AM ne peut prétendre tenir un tel niveau de développement.

- Redondance avec diversité

  Mais il n'est pas possible de s'assurer totalement de l'indépendance entre différents algorithmes d'AM [39]. De plus cela ne permet pas de dépasser le problème de la représentativité des données d'apprentissages et de test.

- Approche fiabilité

  L'analogie avec la fiabilité associée à une défaillance aléatoire n'est pas applicable. La « fiabilité » d'un algorithme d'AM n'est pas aléatoire mais est liée à la représentativité du jeu de test, ou à la robustesse de l'algorithme. Si l'application rencontre un contexte où l'AM généralise mal, alors la probabilité d'occurrence d'un effet adverse est de 1.

- Approche facteur humain

  Il s'agit de l'approche proposée par la NASA avec une licence comme celle des pilotes [4]. Cette approche passe par de nombreux tests, et beaucoup d'expérience. L'humain peut expliquer la raison d'une décision, ce que ne peut pas faire l'AM. C'est une limite insurmontable à ce jour.

- Approche formelle

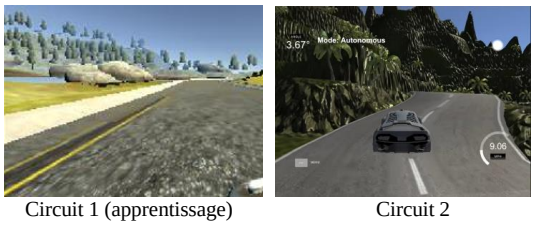
  L'approche formelle permet de démontrer certaines propriétés d'un algorithme. Cette promesse est séduisante : la disponibilité d'outils permet de tester cette approche sur des réseaux de neurones profond ou d'autres algorithmes AM. Les crédits apportés ne semblent pas suffisants pour juger de la sécurité d'un algorithme d'AM au niveau global mais il permet de mieux en comprendre certains aspects.

Aucune de ces approches n'a été jugée adéquate pour justifier d'un niveau de sécurité suffisant pour le cas n°1 (véhicule autonome). C'est pourquoi nous avons retenu ce cas de test.

VIII. PRESENTATION DU CAS TEST

Les éléments issus de ces réflexions sont prévus d'être mis en œuvre sur un modèle simplifié de véhicule autonome à partir des travaux et du logiciel associé [12] [24]. L'approche consiste à définir un angle volant en fonction de 3 prises de vue caméra comme proposé par [24]. Ce modèle dispose d'une piste permettant l'apprentissage et d'une autre piste, différente, pour tester la capacité de généralisation. Ces deux pistes permettent de vérifier si l'algorithme est sûr : soit la voiture peut suivre la nouvelle piste sans sortir de la route, soit elle peut se mettre en sécurité (à l'arrêt sur le côté droit de la route). Elle peut être présentée à un groupe d'analystes afin qu'ils proposent les études ou indicateurs pertinents permettant de garder le véhicule dans un état sûr. Cet exemple à visée pédagogique est encore en développement pour son utilisation en tant que « use case ».

Fig. 14. Extrait de prise de vue des deux circuits présent sur le simulateur



Circuit 1 (apprentissage) Circuit 2

Il est nécessaire de construire un réseau de neurones pour contrôler la trajectoire véhicule. Les travaux [22] ont été repris tels quels. Etant disponibles sur Github en opensource [12], il est possible de les modifier facilement et de les instrumenter. L'approche, extrêmement simplifiée par rapport à un cas réel, consiste à partir des images de caméra à générer une consigne d'angle volant. Pour cela l'auteur utilise un réseau de neurones convolutifs [23]. Les différentes couches sont listées dans la figure 15.

Fig. 15. Description du réseau de neurones

Model: "sequential"

Layer (type)	Output Shape	Param #
lambda (Lambda)	(None, 160, 320, 3)	0
cropping2d (Cropping2D)	(None, 65, 320, 3)	0
conv2d (Conv2D)	(None, 17, 80, 8)	1952
conv2d_1 (Conv2D)	(None, 9, 40, 16)	3216
conv2d_2 (Conv2D)	(None, 9, 40, 32)	8224
flatten (Flatten)	(None, 11520)	0
dropout (Dropout)	(None, 11520)	0
dense (Dense)	(None, 1024)	11797504
dropout_1 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 1)	1025

Le réseau comporte plus de 11 millions de paramètres à calculer.

L'apprentissage se base sur un peu plus de 24 000 images générées à partir de différents tours de circuit effectués par un humain (approche par imitation[18]). 15% des images sont utilisées pour les tests. Afin de multiplier le nombre d'images des transformations sont appliquées (rotation, miroir).

Fig. 16. Vues des 3 caméra (gauche, centrale, droite) du circuit 1

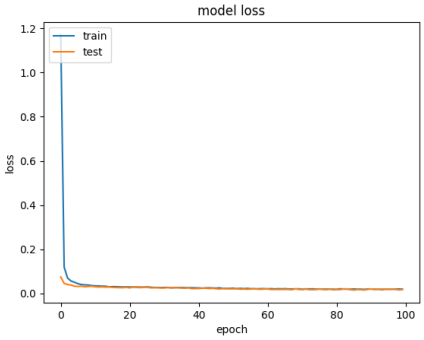
L'algorithme d'apprentissage est l' « adaptive moment estimation » (ADAM) disponible dans KERAS. La fonction de perte « loss function » retenue est la moyenne des carrés des écarts (MSE ou « Mean Square Error »). L'apprentissage est mené sur des batchs de 64 images. Le réseau donne de bonnes performances (i.e. le véhicule effectue un tour du circuit 1 sans accident) à partir de 12 itérations (« epoch » sur Fig. 17). Des tests ont été effectués avec 100 itérations. Dans ce cas, les performances sur le circuit 1 sont similaires, mais le véhicule va un peu plus loin sur le circuit 2. Cela

[18] Il est intéressant de noter que c'est l'approche mise en œuvre pour la compétition MineRL visant à développer une intelligence artificielle gagnant en autonomie [40]. Mais dans cette compétition, cet apprentissage doit être compléter par d'autres approches comme le « reinforcement learning ».

tend à montrer que le réseau a amélioré ses performances en termes de généralisation.

L'apprentissage avec 12 itérations prend 20 minutes sur un PC avec 4Mo de mémoire et un processeur i5. Il s'agit donc d'une machine avec des performances modestes. Il est d'usage de fournir la courbe suivante présentant le taux d'erreurs par génération.

Fig. 17. Fonction de perte (apprentissage et tests)



L'essai se fait sur le même circuit et fonctionne jusqu'à des vitesses de 20km/h. Au-delà de cette vitesse, le véhicule finit par sortir du circuit. Le circuit présente différentes configurations : virages à droite, à gauche, et, des rebords différents.

En revanche, lorsque le véhicule est mis sur l'autre piste, il sort de la route. En fonction des paramètres d'apprentissage, le véhicule peut parcourir quelques centaines de mètres avant de sortir du circuit. C'est une illustration simple des limites de l'algorithme.

Cette expérimentation illustre le besoin de pouvoir déterminer les raisons pour lesquels un réseau échoue par moment dans son comportement observé. Or l'analyse de performance seule ne permet pas d'identifier les causes d'un échec, mais seulement de le constater. Pour en déterminer une partie des causes il est nécessaire d'analyser le comportement du réseau à chaque instant. Des techniques d'analyse statique par interprétation abstraite permettent de calculer les corrélations entre les entrées et les sorties du réseau. Ces corrélations permettent de mettre en valeur l'impact que peut avoir chaque entrée (ici des pixels). L'impact mesuré peut être corrélé positivement (le pixel augmente la valeur de sortie du réseau), ou négativement (le pixel la diminue). Dans cet exemple l'impact n'indique pas de jugement de valeur, car les corrélations positives ou négatives peuvent être tout autant bénéfiques au résultat final. En effet, un angle de volant plus fort n'est pas forcément une mauvaise chose. Dans le cas d'un classifieur, par contre, les corrélations positives renforcent la classification, alors que les négatives conduisent à la diminuer. Pour un classifieur les corrélations positives sont donc plus utiles pour renforcer la robustesse du réseau.

Lors d'une analyse de corrélation a posteriori du réseau on peut représenter les corrélations sur l'image d'entrée à la manière d'une « heatmap » : plus un pixel est bleu plus sa corrélation est négative, plus il est rouge plus elle est positive. Un pixel blanc (transparent) est donc neutre dans le calcul de la sortie.

Fig. 18. Heatmap avec en fond l'image initiale

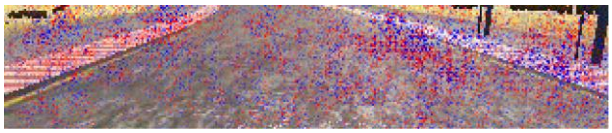
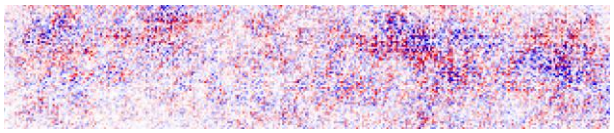


Fig. 19. Heatmap sans l'image initiale



Sur cet exemple le réseau a eu un entrainement très limité. Ce qui explique que les pixels qu'il utilise pour sa prédiction d'orientation du volant sont donc relativement éparpillés. Mais on constate que les bords de la route semblent plus prépondérants que le reste. Il est donc possible que le réseau commence à apprendre que sa réponse devrait être corrélée à la présence ou non des bords de la route sur les côtés de l'image. Il est à noter que ces résultats sont très préliminaires et il est nécessaire encore de travailler l'utilisation que pourrait en faire un algorithme de monitoring. Mais à long terme il est possible d'imaginer des systèmes de monitoring qui vérifient que le réseau analyse des portions de l'image pertinentes.

Il est donc possible de construire le réseau, de le modifier et d'ajouter du monitoring pour tester la mise en place d'un superviseur, comme préconisé dans les travaux du LAAS cités précédemment. Néanmoins il est recommandé de le faire sur le circuit d'apprentissage sans prendre en compte les spécificités de l'autre circuit. Cette démarche montre la difficulté de généraliser une situation lorsqu'il n'est pas possible d'en avoir une connaissance complète.

IX. CONCLUSION ET PERSPECTIVES

La lecture des travaux, et les réflexions sur la base d'une culture plus orientée sûreté de fonctionnement qu'apprentissage machine nous amène à émettre des doutes sur la faisabilité de construire une preuve convaincante de la sécurité d'un réseau de neurones, en l'état actuel de la recherche.

La situation rappelle ce qui s'est passé lorsque le logiciel s'est imposé dans les systèmes. Une démarche s'est dégagée en donnant lieu à la DO178 puis à des approches similaires dans les autres normes sectorielles. La confiance dans la démarche s'est construite sur la durée. Nous arriverons peut-être à une situation identique avec l'apprentissage machine.

Mais nous voyons un obstacle majeur : pour un logiciel « classique », le comportement est prévisible même si cela peut être nuancé pour les logiciels très complexes. Pour l'apprentissage machine, il n'est pas possible à ce jour de s'assurer que la généralisation est correcte et robuste.

Les approches citées dans la littérature ne sont pas mûres, et rien n'indiquent qu'elles le seront rapidement. Les approches actuelles ne permettent pas non plus d'obtenir une confiance suffisante pour une application dont la défaillance peut être catastrophique. Seule l'approche monitoring semble à ce jour apporter une confiance suffisante. Mais cette méthode n'est applicable que pour des cas bien précis où le monitoring, sans avoir les performances de l'AM, peut maintenir le système dans un état acceptable.

Les « heatmap » ou l'approche formelle permettent de comprendre le comportement de l'algorithme localement. Ces deux approches sont doc une aide à la compréhension du comportement de l'algorithme sans avoir de détail sur sa structure.

L'approche « coût / bénéfice » issue du monde médical peut être une solution. Le ratio avantage/risque peut aussi parfois faire basculer le choix vers l'apprentissage machine si ce dernier peut économiser des vies, comme c'est le cas des analyses automatiques de scanners ou les véhicules autonomes. Mais l'acceptabilité sociale des accidents qui se produiront n'est pas acquise. La frilosité de notre société vis-à-vis du risque semble même indiquer que cette acceptabilité sera impossible à obtenir.

Une application, pouvant entraîner des effets catastrophiques, totalement basée sur l'AM est donc exclue. Une percée dans l'apprentissage des algorithmes d'AM est un passage obligé. D'ici là, des mitigations basées sur les spécificités des cas d'usage permettront de déployer l'AM dans les applications critiques (mais sans conséquence catastrophique) et … d'apprendre.

REFERENCES

[1] Frédéric Deschamps, "Driving model and associated safety considérations", https://github.com/iloval98/SAFEIA-AUTONOMOUS -VEHICLE, May 2020 .

[2] Gégoire Savary, Michael Game, Frédéric Deschamps, "Résilience de la fonction Safety lors du développement d'un avion commercial", Lambda Mu 20.

[3] S. C. of RTCA.DO-178C, "Software Considerations in Airborne Systems and Equipment Certification", 2011.

[4] Siddhartha Bhattacharyya, Darren Cofer, David J. Musliner, Joseph Mueller, Eri Engstrom, "Certification Considerations for Adaptive Systems", NASA Langley Research Centern 2015.

[5] Rick Salay, Rodrigo Queiroz, Krzysztof Czarnecki, "An analysis of ISO 26262: Using Machine Learning Safely in Automotive Software", Université de Waterloo, 2017.

[6] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, Dan Mané, "Concrete problems in AI Safety", Google Brain, 2016

[7] Nicolas Rémy, Gabrielle Rives, " Application du Deep Learning dans une architecture HUMS pour la classification du profil d'emploi des systèmes", Lambda Mu 22, 2018.

[8] IRT Saint Exupery, "Projet DEpendable Explainable Learning (DEEL)", https://www.irt-saintexupery.com.

[9] David Gunning, "Program Update November 2017 - Explainable Artificial Intelligence (XAI)", DARPA, 2017.

[10] European Union Aviation Safety Agency (EASA), "Artificial Intelligence Roadmap, human-centric approach to AI in aviation, version 1.0", 2020.

[11] Patrick Hall, Navdeep Gill, "An introduction to Machine Learning Interpretability", O'REILLY, 2018.

[12] https://github.com/udacity/self-driving-car-sim , UDACITY, 2017.

[13] Nicolas Rémy, " Confiance des processus de codécision Homme-IA dans le cadre d'applications de classification supervisée", Lambda Mu 22, 2020.

[14] Guide 51 ISO/CEI:1999, "Aspects liés à la sécurité".

[15] Lola Masson, "Safety monitoring for autonomous systems: interactive elicitation of safety rules", Université Toulouse 3 PaulSabatier (UT3 Paul Sabatier), 2019.

[16] Chiyan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals, "Understanding Deep Learning requires rethinking generalization", Massachussetts Institue of Technology, 2017.

[17] Ian Goodfellow, Yoshua Bengio, Aeron Courville, "Deep Learning", MIT Press, 2016.

[18] Christophe Denis, Franck Varenne, "Inteprétabilité et explicabilité pour l'apprentissage machine : entre modèles descriptifs, modèles prédictifs et modèle causaux. Une nécessaire clarification épistémologique", National conference on Artificial Intelligence (CNIA), 2019.

[19] Vianney Bordeau, François Escudié, "Big data in reliability, Projet de l'IMdR n°P17-4", IMDR, décembre 2019.

[20] INERIS – Rapport d'étude "Guide de mise en œuvre du principe ALARP sur les Installations Classées pour la Protection de l'Environnement (ICPE)", novembre 2014.

[21] Présentation de Numalis ISCLP 2019 à Toulouse.

[22] Thibault Neuveu, https://github.com/thibo73800/self-driving-car , 2018.

[23] lex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Université de Toronto, mai 2017.

[24] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel,Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba, "End to End Learning for Self-Driving Cars", NVIDIA, CoRRabs/1604.07316, 2016

[25] ISO/IEC JTC 1/SC 42 N 478, ISO/IEC NP 24029-2, Artificial Intelligence (AI) -- Assessment of the robustness of neural networks -- Part 2: Formal methods methodology.

[26] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and Harnessing Adversarial Examples", InProc.International Conference on Learning Representations (ICLR), 2015.

[27] Abraham Cherfi, Emmanuel Arbaretier, Linda Zhao et al., "Sécurité-Innocuité des Véhicules Autonomes – Enjeux et Verrous", projet SVA, 2016.

[28] Maël Pégny, Mohamed Issam Ibnouhsein, "Quelle transparence pour les algorithmes d'apprentissage machine ?, Université Paris 1 Panthéon-Sorbonne, 2018.

[29] Gagandeep Singh, Timon Gehr, Markus Püschel, Martin T. Vechev, "An abstract domain for certifying neural networks", Proc. ACM Program. Lang. 3(POPL): 41:1-41:30, 2019.

[30] Zeshan Kurd, Tim Kelly, Jim Austin, "Safety Criteria and Safety Lifecycle for Artificial Neural Networks", University of York.

[31] Alexander Rudolph, StefanVoget, JürgenMottok, "A consistent safety case argumentation for artificial intelligence in safety related automotive systems", ERTS 2018, 2018.

[32] Jens Braband, Hendrik Schäbe, "On Safety Assessment of Artificial Intelligence", Siemens Mobility GmbH, 2020.

[33] https://www.learnopencv.com/understanding-alexnet/

[34] Aurélie Jean, "De l'autre côté de la machine", L'observatoire, 2019.

[35] Thierry Martin, "La probabilité un concept pluriel", Pour la science Hors-série, 2019.

[36] P. Baufreton1, JP. Blanquart, et al., "Multi-domain comparison of safety standards", ERTS, 2010.

[37] Emmanuel Ledinot, Jean-Marc Astruc,et al. "A cross-domain comparison of software development assurance standards", ERTS, 2010.

[38] Jean-Paul Blanquart, Emmanuel Ledinot,et al., "Software Safety – A journey across domainsand safety standards", ERTS, 2018.

[39] Amanda Jane Sharkey, N.E. Sharky, O.C. Copinath, "Diversity, Neural Nets and Safety Critical Applications",University of Sheffield, 1999.

[40] William H. Guss, et al., "NeurIPS 2019 Competition: The MineRL Competition on Sample Efficient Reinforcement Learning using Human Priors", NeurIPS, 2019.