



Actes de la conférence CAID 2021 (Conference on Artificial Intelligence for Defense)

Alexandre Dey, Benjamin Costé, Éric Totel, Adrien Bécue, Elkin Aguas, Anthony Lambert, Gregory Blanc, Hervé Debar, Yannick Chevalier, Amine Medad, et al.

► To cite this version:

Alexandre Dey, Benjamin Costé, Éric Totel, Adrien Bécue, Elkin Aguas, et al.. Actes de la conférence CAID 2021 (Conference on Artificial Intelligence for Defense). CAID 2021 (Conference on Artificial Intelligence for Defense), pp.1-152, 2021. hal-03535661

HAL Id: hal-03535661

<https://hal.science/hal-03535661>

Submitted on 6 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Actes de la conférence CAID 2021

(Conference on Artificial Intelligence for Defense)

Organisée par



Intelligence artificielle appliquée à la sécurité des réseaux et des systèmes d'information

Simulation réaliste d'utilisateurs pour les systèmes d'information en Cyber Range

Alexandre Dey, Benjamin Costé, Eric Totel et Adrien Bécue

Automated Botnet Traffic Filtering Using Deep Reinforcement Learning

Elkin Aguas, Anthony Lambert, Grégory Blanc and Hervé Debar

Data Exchange for Anomaly Detection: Analysis of CAN bus logs

Yannick Chevalier

Real-time graph clustering for network intrusion detection

Amine Medad, Baptiste Gregorutti, Edouard Genetay and Alexandre Peter Nguema

Désobfuscation par intelligence artificielle : comment s'en protéger?

Grégoire Menguy, Sebastien Bardin, Richard Bonichon et Cauim De Souza Lima

IA appliquée aux systèmes critiques

Multi-fidelity constrained Bayesian optimization, application to drone design

Rémy Charayron, Thierry Lefèbvre, Nathalie Bartoli and Joseph Morlier

Formal Methods for AI: Lessons from the past, promises of the future

Zakaria Chihani

The benefits of using ALTAI in Defence applications

Bruno Carron and Stephan Brunessaux

Aeronautical ad-hoc networking based on artificial intelligence minimizing interference on ground networks

Léonard Caquot, Tristan Charrier, Badre El Bezzaz Semlali, Oudomsack Pierre Pasquero, and Alexis Bazin

Cryptographie appliquée à l'apprentissage machine distribué

MPC4SaferLearn: Privacy-Preserving Collaborative Learning with Secure and Robust Decentralized Aggregation

Katarzyna Kapusta and Pierre-Elisée Flory

PRIVILEGE: PRIVacy and Homomorphic Encryption for Artificial Intelligence

Oana Stan, Vincent Thouvenot, Karel Hynek, Romain Ferrari, Aymen Boudguiga, Renaud Sirdey, Alice Heliou, Tomas Cejka, Katarzyna Kapusta, Daria La Rocca, Martin Zuber, George Vardoulas, Ioannis Papaioannou, Andreas Vekinis and Georgia Papadopoulou

Classification à base d'IA

Monte Carlo Tree Search for Multi-function Radar Task Scheduling

Marc Vincent, Amal El Fallah Seghrouchni, Vincent Corruble, Narayan Bernardin, Rami Kassab and Frédéric Barbaresco

Évaluation statistique efficace de la robustesse de classifieurs

Karim Tit, Teddy Furon, Mathias Rousset and Louis-Marie Traonouez

Generalized $SU(1; 1)$ Equivariant Convolution on Fock-Bargmann Spaces for Robust Radar Doppler Signal Classification

Pierre-Yves Lagrave and Frederic Barbaresco

Intelligence artificielle pour l'imagerie en temps réel

Case-based reasoning for rare events prediction on strategic sites

Vincent Vidal, Marie-Caroline Corbineau and Tugdual Ceillier

End-to-End Pipeline for Visually Rich Documents Comprehension using Deep Learning applied to Nuclear Industry?

Aleksei Iancheruk, Ahmed Allali, Julien Rodriguez, Tejas Bhor, Ricardo Garcia, Jean-Eudes Guilhot-Gaudeffroy and Robert Plana

Simulation réaliste d'utilisateurs pour les systèmes d'information en Cyber Range

Alexandre Dey^{1,2}, Benjamin Costé¹, Éric Totel³ et Adrien Bécue¹

¹ Airbus CyberSecurity, Rennes, France

² IRISA, Rennes, France

³ Télécom SudParis, Paris, France

`alexandre.dey@airbus.com`

`benjamin.b.coste@airbus.com`

`eric.totel@telecom-sudparis.eu`

`adrien.becue@airbus.com`

Résumé. La génération d'activité utilisateur est un élément-clé autant pour la qualification des produits de supervision de sécurité que pour la crédibilité des environnements d'analyse de l'attaquant. Ce travail aborde la génération automatique d'une telle activité en instrumentant chaque poste utilisateur à l'aide d'un agent externe; lequel combine des méthodes déterministes et d'apprentissage profond, qui le rendent adaptable à différents environnements, sans pour autant dégrader ses performances. La préparation de scénarios de vie cohérents à l'échelle du SI est assistée par des modèles de génération de conversations et de documents crédibles.

Mots-clé: Cyber range · génération de texte · reconnaissance d'images · simulation de vie · jeux de données · honeynet

1 Introduction

Les avancées récentes et la démocratisation des technologies de virtualisation (e.g., *Software Defined Network* ou SDN, Cloud) ont notamment permis l'essor d'outils dédiés au cyber-entraînement. Appelées communément Cyber Ranges, ces plateformes facilitent le déploiement de Systèmes d'Information (SI) complets pour l'organisation de formations et exercices à destination des opérationnels de cyberdéfense. Outre leurs qualités pédagogiques, elles constituent également une base solide pour l'évaluation et la mise au point des outils de supervision de sécurité [9] (sonde de détection d'intrusion, détection d'anomalies, SIEM, etc.), ainsi que la constitution d'environnements d'analyse des attaquants (*honeynet* et plateformes de détonation).

La simulation d'activité sur les postes utilisateur apporte une crédibilité nécessaire aux plateformes d'analyse des attaquants tout en permettant d'évaluer le comportement des produits de sécurité face au fonctionnement nominal des SI. En effet, les jeux de données disponibles pour l'apprentissage machine dans le domaine de la cybersécurité représentent des attaques, plus ou moins réalistes,

mais n'intègrent pas ou peu de comportement illégitimes (scan de ports, branchement de clés USB sur des postes sensibles, etc.) émanant d'activités d'utilisateur légitimes. Cette absence d'activité, pourtant foisonnante sur un SI réel, complexifie l'immersion dans le cas du cyber-entraînement, biaise les données collectées sur les terminaux pour l'entraînement de méthodes de supervision basées sur l'apprentissage machine, et diminue grandement la crédibilité des plateformes d'analyse des attaquants.

La génération automatique de vie réaliste, objet notamment du challenge IA & Cyber 2020 de l'ECW, est un sujet complexe qui demande de résoudre plusieurs problèmes. Tout d'abord, l'instrumentation des machines doit se faire par des méthodes extérieures à celles-ci afin de limiter les traces de simulation laissées sur les postes utilisateurs. En second lieu, il est nécessaire d'adapter en temps réel le scénario pré-établi aux réactions aléatoires de l'environnement de simulation (e.g., position d'une fenêtre, arrêt imprévu). Cette adaptation se fait via des vérifications automatiques de l'environnement qui doivent par ailleurs être effectuées en un temps inférieur ou égal au temps de réaction humain. Enfin, une assistance à l'opérateur est nécessaire pour la mise au point de scénarios à grande échelle.

Ce travail s'inspire des méthodes de génération de vie, reposant sur un agent, employées par la plateforme de détonation BEEZH, présentée par Amossys à la conférence C&ESAR 2020 [7], pour lesquelles nous proposons plusieurs améliorations :

- Découpage de l'agent en plusieurs couches d'abstraction successives pour gagner en modularité (e.g., s'affranchir de la technologie de virtualisation, adaptabilité à des machines physiques, etc.);
- Combinaison efficiente de méthodes déterministes et d'apprentissage profond pour la réaction en temps réel de l'agent;
- Assistance à la création d'actions exécutables par l'agent;
- Assistance à la création de scénarios de vie à partir des profils des utilisateurs simulés et de leurs interactions.

Dans la section 2, nous présentons l'état de l'art. Les sections 3 et 4 décriront les travaux réalisés. Nous concluons en discutant notre approche en sections 5 et 6.

2 État de l'art

De nombreuses études ont souligné l'importance de la simulation d'activité utilisateur pour l'étude des logiciels malveillants [4, 1]. Ainsi, l'absence de plusieurs artefacts (e.g., présence de fichiers dans la corbeille, présence de cookies, historique de navigation) permet d'identifier les machines n'ayant jamais été utilisées [15]. Certains logiciels malveillants contournent les systèmes de détection par le biais de *tests de Turing inversés* tels que des mécanismes à retardement (scroll dans un document Word) ou la vérification de la vitesse d'utilisation de la souris

[25]. La simulation en temps réel d'actions utilisateurs permet cependant de mettre en évidence le comportement malveillants de ces logiciels.

L'utilisation automatisée de l'interface graphique est un sujet historiquement étudié pour le contrôle qualité des logiciels. Deux approches s'y distinguent : le rejeu d'actions pré-enregistrées [27, 24, 16], long à configurer, et l'automatisation complète qui souffre d'un manque de réalisme [23, 6]. Chacune de ces méthodes requiert un agent installé sur les machines instrumentées, ce qui fournit un indicateur à l'attaquant, et teinte les journaux d'événements dans le cas de la collecte de jeux de données.

Plus récemment, MORRIGU [14] instrumente des machines virtuelles via l'API VirtualBox depuis un hôte Windows. Malgré ses résultats positifs sur la détection de comportements malveillants, le manque de portabilité de la solution ne permet pas d'envisager son emploi à des fins de cyber-entraînement qui implique l'instrumentation de multiples machines en réseau.

L'outil de simulation des utilisateurs intégré à la plateforme BEEZH [7] instrumente des machines virtuelles au travers de la fonction de déport d'écran de l'hyperviseur, reposant sur la technologie VNC. Les scénarios de vie sont découpés en actions unitaires simples (e.g., ouvrir le navigateur, recherche web) dont l'exécution est assistée par des méthodes d'analyse d'images (captures d'écran) à l'aide de la librairie *desker* [2], publiée sous licence GPL v3. Cette approche pionnière a montré l'importance mais également la complexité du sujet. Le recours à VNC offre en effet un large éventail de possibilités vis-à-vis de l'interaction avec la machine instrumentée mais se restreint néanmoins à l'instrumentation de machines virtuelles. La création des scénarios et leur adaptation aux spécificités de l'environnement nécessitent de surcroît des actions entièrement manuelles. Enfin, la solution d'analyse d'image retenue (Faster-RCNN [22]) sollicite d'importantes ressources de calcul incompatibles avec notre contrainte de performance. Faster-RCNN [22] est un algorithme très répandu (car pertinent) pour la détection de zones d'intérêt. D'autres initiatives plus récentes telles que RetinaNet [12] ou SSD [13] fournissent cependant de meilleurs résultats en un temps plus court.

La génération automatique de texte repose fréquemment sur les modèles de Markov cachés [8] (HMM, *Hidden Markov Model*) mais les résultats produits deviennent rapidement incohérents lorsque la séquence générée s'allonge. Des solutions à base de réseaux de neurones comme Transformer [26] apparaissent plus adaptés à la modélisation du langage. Les modèles basés sur le système GPT [18], performants et polyvalents, produisent toutefois des modèles massifs (e.g., 11 milliards de paramètres pour T5 [20], 175 milliards pour GPT-3 [3]). Au regard de leurs performances, nos travaux s'inspirent de GPT-2 [19] (117 millions de paramètres) et CTRL [10] pour la génération conditionnelle de texte.

3 Exécution et enregistrement d'actions

L'instrumentation de machines virtuelles ou physiques repose sur la conception d'un agent externe (Section 3.1) capable de reconnaître son environnement et

de s'y adapter en conséquence (Section 3.2). La réalisation d'actions complexes pré-enregistrées (Section 3.3) est également une fonctionnalité recherchée.

3.1 Conception de l'agent

Afin de simplifier la configuration du générateur de vie par les opérateurs, nous avons choisi une approche en couches d'abstraction (Fig. 1). En effet, une approche de bout-en-bout devient difficile à maintenir le nombre de scénarios grandissant (e.g., code dupliqué) et impose à l'opérateur de définir manuellement un grand nombre de détails lorsqu'il crée un scénario. Les scénarios d'activité utilisateur sont ainsi transcrits par un agent en actions bas niveau (clavier, souris et écran) qu'il effectue ensuite sur la machine instrumentée.

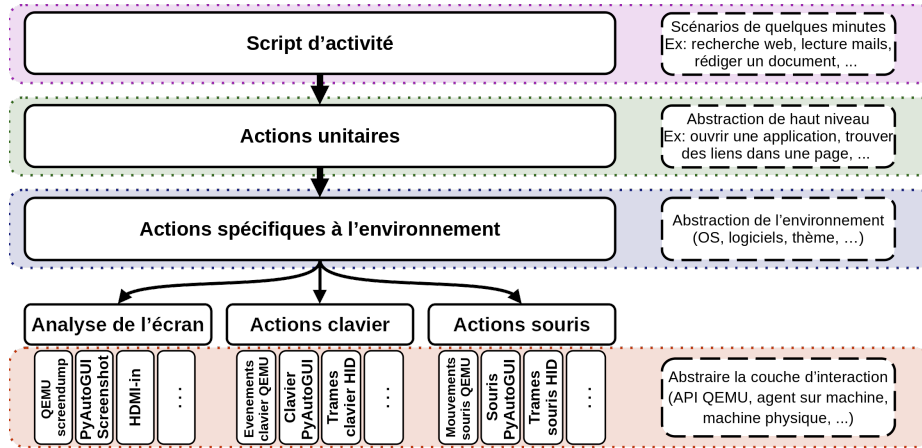


Fig. 1. Architecture fonctionnelle de l'agent

Au plus bas niveau, l'utilisateur virtuel interagit avec la machine instrumentée par le biais de la souris, du clavier et de l'écran. L'objectif est de pouvoir, sans modifier les scénarios de vie, adapter l'agent à plusieurs plateformes de virtualisation (e.g. VMWare ESXi, Proxmox, etc.) mais également à des machines physiques. Nous avons considéré pour cela trois méthodes d'interaction différentes mais complémentaires : l'API Qemu Monitor⁴, une connexion VNC et un agent installé sur la machine à instrumenter. Concernant les machines physiques, l'approche la plus pertinente semble la communication des actions clavier et souris par USB selon le protocole HID⁵ en capturant la sortie vidéo.

Pour contrer des méthodes d'évasion de sandbox employées par certains logiciels malveillants, nous rajoutons de l'aléa dans les frappes clavier et les mouvements de souris. Pour ces derniers, chaque déplacement est découpé en une

⁴ <https://qemu-project.gitlab.io/qemu/system/monitor.html>

⁵ <https://www.usb.org/hid>

multitude de petits déplacements, et la vitesse varie en fonction de la distance à parcourir (i.e., un mouvement court sera moins rapide qu'un mouvement long). Ceci fluidifie le mouvement (augmentant le réalisme) et trompe les malware qui détectent les déplacements instantanés de la souris. De l'aléa et une inertie dans le mouvement sont rajoutés pour éviter les mouvements de souris rectilignes et trop précis (détectés par certaines méthodes d'évasion). Une latence aléatoire est également ajoutée entre chaque frappe clavier afin de rendre inopérantes les méthodes de détection recherchant une régularité. Enfin, lorsque l'utilisateur virtuel est en attente (e.g., lors de la lecture d'un texte, ou lorsque l'agent effectue un calcul long), des mouvements aléatoires de souris sont effectués pour simuler les mouvements spontanés lorsque la main est posée sur la souris. Les modèles d'aléa présentés ici visent à contrer les mécanismes de sécurité les plus simples. Ces modèles pourront être complexifiés par la suite afin de modéliser plus finement des utilisateurs humains (e.g., vitesse de frappe différente d'un utilisateur à un autre).

La seconde couche dite d'interaction permet de s'adapter aux spécificités de l'environnement de la machine instrumentée. En effet, bien que les méthodes de reconnaissance d'images utilisées (section 3.2) soient peu sensibles aux variations graphiques mineures (e.g., résolutions d'écran différentes, couleurs légèrement différentes, etc.), certaines variations demandent de modifier les images à cibler. Par exemple, les actions à effectuer pour envoyer un mail avec Thunderbird ou Outlook sont fonctionnellement similaires mais les éléments d'interface sont suffisamment différents pour nécessiter des interactions différentes avec la machine.

La couche supérieure décrit des actions unitaires simples (e.g., ouvrir le navigateur web, rechercher un mot clé, identifier les liens dans du texte, etc.) et sert d'interface de programmation (API) pour les scénarios. Les scénarios ainsi créés constituent l'ultime couche d'abstraction.

3.2 Analyse des captures d'écran

L'analyse des captures d'écran permet à un agent de contrôler son état en temps réel par reconnaissance des zones d'intérêts, telles les boutons d'interface à cliquer, les liens dans une page web, etc. Pour garantir l'efficacité calculatoire, primordiale dans notre contexte, nous limitons systématiquement la quantité d'information (i.e. taille et nombre d'images) et la complexité des modèles statistiques. Trois techniques de reconnaissance d'image sont ainsi employées. La première, connue sous le nom de *template matching*, consiste à trouver dans une image (ici une capture d'écran) le ou les éléments correspondant le plus à une cible recherchée. Bien que de faibles variations (e.g., résolution, couleurs, etc.) suffisent à la rendre inopérante, cette méthode a l'avantage d'être peu coûteuse en ressources et d'être déterministe, ce qui la rend appropriée à la détection d'éléments d'interface qui varient peu voire pas pour un même environnement (e.g., bouton du menu démarrer, des fenêtres, etc.).

Dans les cas où le *template matching* n'est pas satisfaisant, nous proposons d'employer des algorithmes d'apprentissage profond ayant montré des performances remarquables pour la détection d'objet. Les méthodes les plus efficaces

telles SSD [13] ou YOLO [21] requièrent cependant d'importantes ressources calculatoires qui limitent le passage à l'échelle (plusieurs machines instrumentées). De plus, il n'existe pas de jeux de données publics contenant des captures d'écran avec les zones d'intérêt détournées et annotées. Collecter et annoter un tel jeu de données prend un temps considérable. Pour ces raisons, nous avons choisi une approche qui exploite la géométrie des formes à repérer à l'écran (Fig. 2) en appliquant une méthode de seuillage adaptatif [11]. Les objets au premier plan (e.g., icônes, texte) ainsi que les séparations entre les différentes zones à l'écran (e.g., contour des fenêtres) deviennent ainsi simples à identifier. En appliquant des règles de filtrage (e.g., une icône est à peu près aussi large que haute, un bouton sera plutôt plus large que haut, etc.) il est possible, pour un coût calculatoire faible, d'identifier les zones d'intérêt. Des modèles statistiques (e.g., réseaux de neurones à convolution) filtrent les potentiels faux positifs et classifient les autres.

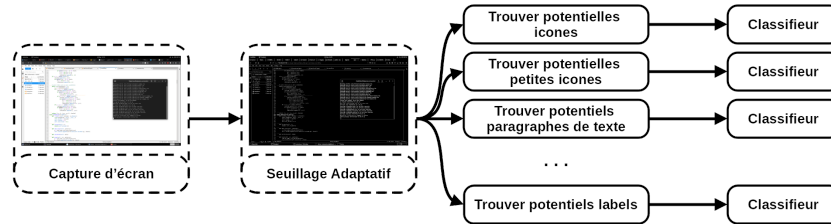


Fig. 2. Méthodologie d'analyse d'images

Enfin, la reconnaissance de caractères est utilisée pour la navigation dans l'interface (e.g., label des fichiers et dossiers, éléments des menus, etc.) ainsi que la reconnaissance des liens dans les pages web. Notre implémentation utilise la librairie tesseract [17] pour l'analyse des éléments textuels de l'interface, et reprend la même méthode que desker [2] pour la détection de liens.

3.3 Aide à la création d'actions

La reconnaissance d'image présentée dans la section 3.2 donne à l'agent la capacité de reconnaître l'environnement dans lequel il évolue et s'y adapter. Par exemple, l'interface de Firefox diffère de celle de Chrome, bien que les fonctionnalités de ces navigateurs soient équivalentes. Par conséquent, l'agent cherchera des repères visuels différents pour les manipuler. Nous proposons un outil de création d'actions qui exploite ces similarités pour faciliter la collecte de données destinées aux techniques de reconnaissance d'images présentées plus haut.

Cet outil enregistre les mouvements et clics de la souris, les frappes du clavier ainsi que des captures d'écran. Ces données brutes sont ensuite agrégées pour découper l'enregistrement en petites actions (e.g., mouvement de la souris de A vers B, puis double clic gauche sur B). Les zones d'intérêts dans les captures

d'écran sont extraites selon deux méthodes différentes. La première consiste à comparer une image prise au moment d'un clic de souris avec l'image prise au début du mouvement précédent ce clic. La plupart des interfaces graphiques actuelles mettent en surbrillance les éléments survolés par la souris ce qui permet d'extraire simplement ces éléments, qui serviront de cibles de *pattern matching*. La seconde méthode consiste à employer la technique d'identification des zones potentiellement intéressantes décrite dans la section 3.2 (Fig. 2) ce qui accélère grandement l'annotation des jeux de données pour l'entraînement de modèles de reconnaissance de zones d'intérêt.

4 Gestion de scénarios de vie à l'échelle du système

Les outils et méthodes présentées permettent à notre agent d'interagir et de s'adapter à son environnement. La présente section aborde la réalisation et l'exécution de scénarios de complexités variées à l'échelle d'un SI comportant plusieurs machines et utilisateurs.

4.1 Orchestration des communications entre utilisateurs

Dans le contexte du cyber-entraînement, pour améliorer la cohérence et simplifier la mise en place de l'exercice, il est intéressant de créer des avatars pour chacun des utilisateurs simulés et de leurs interactions. En fonction de leurs rôles dans l'entreprise et des relations qu'ils entretiennent avec leurs collègues, les employés vont interagir différemment avec le système d'information simulé. Par exemple, deux collègues amis en dehors de leur travail sont plus susceptibles de discuter par mail de manière informelle que deux collègues ne se connaissant pas. Similairement, un développeur utilisera plus souvent l'éditeur de code qu'un manager.

Nous générons des canevas de scénarios de vie à l'échelle du système par le biais d'un graphe relationnel des utilisateurs de ce système (Fig. 3). Ce graphe est composé des différents avatars, des projets sur lesquels ils travaillent, de leurs groupes de travail et de la nature des relations qu'ils entretiennent entre eux (e.g., amicales, client-fournisseur, partenaires, lien hiérarchique, etc.).

4.2 Modèles de génération de texte

L'ajout de contenu réaliste renforce notablement la crédibilité de la vie simulée. Notamment, l'absence d'échanges de mails ou de documents sur les postes utilisateur est un indicateur fort pour un adversaire, y compris si le contenu des mails et des documents est manifestement non crédible (e.g., succession de mots vides de sens). Cependant, la génération manuelle de ce contenu est laborieuse. En effet, rédiger du texte à la manière d'un avatar est en soi une tâche fastidieuse qu'il est pourtant nécessaire de reproduire pour plusieurs dizaines d'avatars dans le cas de scénarios complexes.

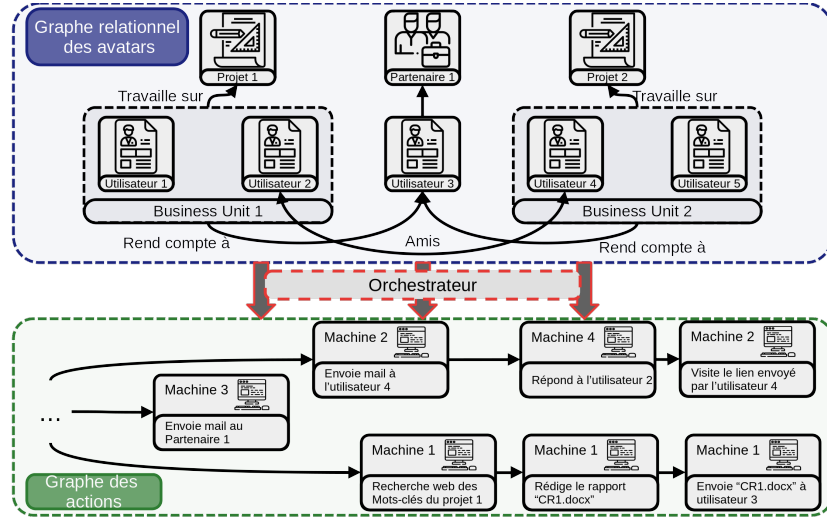


Fig. 3. Vue d'ensemble de l'orchestrateur

Le domaine du traitement du langage naturel (NLP, *Natural Language Processing*) a récemment connu une avancée majeure avec la démocratisation des réseaux de neurones type Transformer [26]. Le mécanisme d'attention, principale particularité de ces modèles, leur permet de modéliser avec précision la syntaxe et la sémantique de plusieurs langages naturels. OpenAI a démontré l'efficacité de l'apprentissage par transfert avec des modèles comme GPT [18], ainsi que ses capacités dans le domaine de la génération de texte. Le modèle employé génère du texte dans la continuité d'un contexte fourni au préalable (souvent des mots commençant une phrase). Dans notre cas, nous affinons le modèle pré-entraîné GPT-2 [19] pour la génération conditionnelle de texte. Plus spécifiquement, le contexte fourni en entrée du modèle contient des informations sur le ton à employer dans le texte (e.g., professionnel, informel, à un partenaire, à un client) ainsi que les thématiques à aborder (e.g., le sujet du mail, des mots clés).

Pour vérifier l'efficacité de notre méthode de génération de texte, nous avons entraîné un modèle sur un jeu de donnée de 35 000 mails récupérés en source ouverte. Ces mails sont associés à un ensemble de mots-clés, une polarité (positif, négatif ou neutre) et une tonalité (formel, informel).

5 Discussion

5.1 Actions exécutées par l'agent

Afin de ne pas laisser de traces d'instrumentation sur la machine cible, nous avons développé un agent qui communique au travers d'interfaces externes (clavier, souris, écran), le rendant ainsi indétectable pour un attaquant ou un outil de supervision (e.g., EDR). L'agent développé est apte à effectuer des activités de

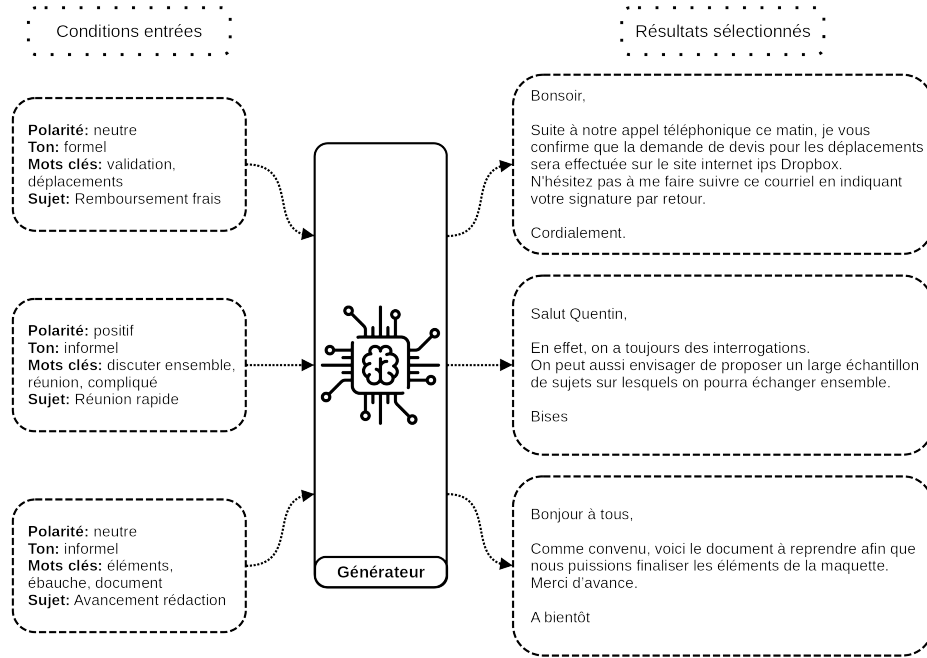


Fig. 4. Exemple de mails générés conditionnellement avec le modèle dérivé de GPT-2

bureautique simples (lecture/écriture de mail, navigation web, traitement de texte, etc.). L'automatisation est de plus masquée par l'ajout d'un aléa dans les mouvements de souris et les frappes clavier qui compliquent la détection automatique de cette instrumentation (ce que font certains logiciels malveillants pour détecter des sandbox [25]), obligeant ainsi un attaquant à évaluer la crédibilité du scénario en lui même.

Le panel des actions que l'agent peut réaliser améliore significativement la robustesse des systèmes actuels face à des malware utilisant des tests de Turing inversés. En revanche, la bibliothèque d'actions devra être complétée pour créer des jeux de données réalistes destinés à valider le fonctionnement d'algorithmes de détection. En particulier, la réalisation de tâches d'administration et plus largement des actions générant du bruit au niveau des outils de supervision (e.g., un administrateur ne respectant pas les procédures, un utilisateur utilisant des outils de scan réseau, etc.) permettraient de différencier les niveaux techniques des utilisateurs simulés et, par la même occasion, d'enrichir les jeux de données pour la détection.

L'enregistreur d'actions permet d'adapter en quelques minutes les actions existantes à des environnements nouveaux. Ceci permet d'une part d'intégrer rapidement, sans connaissances préalables sur le fonctionnement de notre outil, de nouvelles actions supportant des cas non gérés (e.g., une nouvelle pop-up pour la gestion des cookies lors de la navigation web, une mise à jour majeure d'un

logiciel, etc.) et d'autre part, de réaliser des actions complexes qui nécessiteraient un temps de développement important. L'adaptation de scénarios complets, bien que grandement facilitée, demeure toutefois une activité complexe.

5.2 Génération de scénarios

Ce travail présente la conception d'un agent seul et ne traite donc pas de l'orchestration des agents au sein d'un environnement multi-machines. Cette problématique comprend la communication et l'organisation entre les agents (e.g., répondre à mail) mais également la gestion de profils utilisateurs brièvement abordée dans la Section 4.1.

De plus, le modèle de génération de texte actuellement développé génère une majorité de résultats non crédibles : environ 20% sont exploitables avec des modifications mineures. Ceci nous empêche de générer la totalité des échanges sans intervention humaine. Bien que la génération ne soit pas entièrement automatique, celle-ci est toutefois grandement facilitée. En effet, il est possible, à partir du même contexte, de générer plusieurs candidats que l'opérateur peut utiliser pour gagner du temps dans la génération des mails et documents du scénario. Nous envisageons deux axes majeurs pour améliorer ces capacités de génération :

1. Augmenter la quantité de données et la taille du modèle (e.g., utiliser une autre variante de GPT-2);
2. Rajouter une étape d'entraînement type GAN (Generative Adversarial Network) [5], pour inciter le modèle à générer des candidats plus réalistes.

6 Conclusion

Nous avons présenté une méthode pour simuler des utilisateurs à l'échelle d'un SI complet en favorisant l'adaptabilité et la simplicité de mise en œuvre. Notre approche emploie un agent externe découpé en plusieurs niveaux d'abstraction, avec, au plus bas, une interaction avec les machines instrumentées au travers du clavier, de la souris et de l'écran. Pour faire face à la diversité des interfaces utilisateur des systèmes modernes (e.g., OS différents, logiciels différents, etc.) l'agent embarque des techniques de reconnaissance d'images reposant à la fois sur des méthodes déterministes (template matching) et d'apprentissage profond (réseaux de neurones à convolution). Ceci équilibre la quantité de ressources calculatoires requises par l'agent avec la flexibilité offerte par les méthodes probabilistes. Un enregistreur d'actions simplifie la configuration de l'agent pour tout nouvel environnement en extrayant automatiquement les images cibles, et en facilitant la collecte et l'annotation de données d'entraînement pour les modèles statistiques. La création de scénarios de vie se base sur les avatars des utilisateurs virtuels et de leurs interactions dont se nourrissent nos modèles de génération conditionnelle de texte qui produisent en masse des conversations e-mail réalistes et des documents crédibles.

Notre proposition enrichit la méthode de génération de vie initialement proposée pour la plateforme BEEZH par l'amélioration des performances et le

réalisme de l'activité générée. Nous complétons également la proposition initiale par une assistance pour la génération de scénarios à grande échelle. En cours d'implémentation, nos travaux bénéficient de premiers résultats encourageant qui restent toutefois à consolider avant de valider expérimentalement notre proposition.

Acknowledgments Les auteurs remercient chaleureusement Alexandre De Beaudrap dont les travaux de stage ont permis de consolider l'approche.

References

1. Afianian, A., Niksefat, S., Sadeghiyan, B., Baptiste, D.: Malware Dynamic Analysis Evasion Techniques: A Survey. arXiv:1811.01190 [cs] (Nov 2018), <http://arxiv.org/abs/1811.01190>, arXiv: 1811.01190
2. Amossys: deser. <https://gitlab.com/d3sker/deser> (2021)
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Nee-lakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
4. Bulazel, A., Yener, B.: A Survey On Automated Dynamic Malware Analysis Evasion and Counter-Evasion: PC, Mobile, and Web. In: Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium. pp. 1–21. ROOTS, Association for Computing Machinery, Vienna, Austria (nov 2017). <https://doi.org/10.1145/3150376.3150378>
5. Croce, D., Castellucci, G., Basili, R.: GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2114–2119. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.191>, <https://aclanthology.org/2020.acl-main.191>
6. Feng, P., Sun, J., Liu, S., Sun, K.: UBER: Combating Sandbox Evasion via User Behavior Emulators. In: Zhou, J., Luo, X., Shen, Q., Xu, Z. (eds.) Information and Communications Security, vol. 11999, pp. 34–50. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-41579-2_3
7. Guihéry, F., Siffer, A., Paillard, J.: Beezh: une plateforme de détonation réaliste pour l'analyse des modes opératoires d'attaquants (2020)
8. Jelinek, F.: Markov source modeling of text generation. In: The impact of processing techniques on communications, pp. 569–591. Springer (1985)
9. Jiang, H., Choi, T., Ko, R.K.: Pandora: A cyber range environment for the safe testing and deployment of autonomous cyber attack tools. arXiv preprint arXiv:2009.11484 (2020)
10. Keskar, N.S., McCann, B., Varshney, L.R., Xiong, C., Socher, R.: CTRL: A conditional transformer language model for controllable generation (2019)
11. Lie, W.N.: Automatic target segmentation by locally adaptive image thresholding. IEEE Transactions on Image Processing **4**(7), 1036–1041 (1995)
12. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection (2018)

13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. *Lecture Notes in Computer Science* pp. 21–37 (2016)
14. Mills, A., Legg, P.: Investigating Anti-Evasion Malware Triggers Using Automated Sandbox Reconfiguration Techniques. *Journal of Cybersecurity and Privacy* **1**(1), 19–39 (Mar 2021). <https://doi.org/10.3390/jcp1010003>, <https://www.mdpi.com/2624-800X/1/1/3>
15. Miramirkhani, N., Appini, M.P., Nikiforakis, N., Polychronakis, M.: Spotless Sandboxes: Evading Malware Analysis Systems Using Wear-and-Tear Artifacts. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 1009–1024. IEEE, San Jose, CA, USA (May 2017). <https://doi.org/10.1109/SP.2017.42>, <http://ieeexplore.ieee.org/document/7958622/>
16. Nguyen, B.N., Robbins, B., Banerjee, I., Memon, A.: Guitar: an innovative tool for automated testing of gui-driven software. *Automated software engineering* **21**(1), 65–105 (2014)
17. tesseract ocr: tesseract. <https://github.com/tesseract-ocr/tesseract> (2019)
18. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
19. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9 (2019)
20. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection (2016)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks (2016)
23. Rueda, U., Vos, T.E., Almenar, F., Martínez, M., Esparcia-Alcázar, A.I.: Testar: from academic prototype towards an industry-ready tool for automated testing at the user interface level. *Actas de las XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2015)* pp. 236–245 (2015)
24. Singhera, Z., Horowitz, E., Shah, A.: A graphical user interface (gui) testing methodology. *International Journal of Information Technology and Web Engineering (IJITWE)* **3**(2), 1–18 (2008)
25. Vashisht, S.O., Singh, A.: Turing Test in Reverse: New Sandbox-Evasion Techniques Seek Human Interaction (2014), <https://www.fireeye.com/blog/threat-research/2014/06/turing-test-in-reverse-new-sandbox-evasion-techniques-seek-human-interaction.html>
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Advances in neural information processing systems*. pp. 5998–6008 (2017)
27. Yeh, T., Chang, T.H., Miller, R.C.: Sikuli: using gui screenshots for search and automation. In: *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. pp. 183–192 (2009)

Automated Botnet Traffic Filtering Using Deep Reinforcement Learning

Elkin Aguas^{1,2}, Anthony Lambert², Grégory Blanc¹, and Hervé Debar¹

¹ Télécom SudParis, Institut Polytechnique de Paris, 9 Rue Charles Fourier, 91000 Evry-Courcouronnes, France

{name.surname}@telecom-sudparis.eu

² Orange Labs, 46 Avenue de la République, 92320 Châtillon, France

{name.surname}@orange.com

Abstract. The rising popularity of network automation has sparked a transition from error-prone, time-consuming manual manipulations to agile and refined automated orchestration aimed at improving network management and security. This paper investigates the capabilities of a deep reinforcement learning agent to learn how to automatically adapt filtering actions towards seemingly botnet-originated traffic. Our work focuses on early mitigation of Mirai-like IoT botnets in an ISP core network. Our contribution is threefold. We propose a realistic Mirai-like botnet traffic model, a new architecture to emulate generic network infrastructures, and a Deep Reinforcement Learning agent to control the automated mitigation of botnets. Preliminary results suggest that agent-controlled botnet mitigation automation is feasible.

Keywords: deep reinforcement learning · network · automation · security · management · botnet.

1 Introduction

Botnets are a fundamental part of cyber criminality in the modern Internet. Their constant growth [1, 2], combined with their essential role in malware propagation throughout the Internet, makes them one of the most harmful threats cybersecurity experts and network operators face today.

A botnet consists of three fundamental elements: bots (i.e., the compromised machines), a command and control (C&C) server (i.e., the communication channel), and a botmaster (i.e., the attacker). Knowing these elements, we can define a botnet as a group of compromised machines under the direct or indirect control of an attacker through a secure remote channel. Botnet families differ in various aspects, such as the protocols used to communicate, their objectives, and spreading vectors. However, based on their topology, there are two major botnet types: centralized and decentralized botnets.

Modern decentralized botnets have an identifiable life cycle that helps to trace propagation and infection patterns. Different life cycle taxonomies have been proposed based on a variety of aspects, such as financial, functional, and

virological [3–5] ones. These taxonomies often present certain stages common to a generic decentralized botnet life cycle: for instance, Khattak et al. [6]’s botnet behavior-based taxonomy identifies 5 life cycle related features, which are *propagation*, *rallying mechanisms* (i.e., how to register to the botnet), *C&C*, *purpose* and *evasion*.

Knowing that botnets can be characterized by their life cycle is essential for early detection and mitigation, especially during their propagation and infection stages. This is of significant interest for Internet Service Providers (ISP), because early botnet detection leads to early mitigation, which disrupts botnet propagation and growth, and can reduce the effect and even stop ulterior attacks. Furthermore, the automation of botnet early mitigation should be a key concern for ISPs, since automation can make the process more economically efficient and scalable [7].

There is a wide range of cyber attacks that use botnets as their base infrastructure. In many cases, the harm a botnet can cause is directly proportional to its size. This is particularly relevant for attacks such as Distributed Denial of Service (DDoS) and information leakage [8]. Nonetheless, botnets can be used for attacks that require more complex actions, such as cryptocurrency mining [9]. Other attacks that can be equally harmful for ISPs and companies are click fraud, identity theft, scareware, keylogging, traffic sniffing [10, 11]. This paper evaluates the capabilities of Deep Reinforcement Learning (DRL) to control the automated deployment of filtering actions in a router, and how this can mitigate botnet traffic in a network.

This article is organized as follows: Section 2 describes the problematic and use case. Section 4 introduces our solution and presents initial results. Section 5 concludes the paper.

2 Problem Statement and Use Case Description

Botnets can be used to directly attack ISPs or companies and services in an ISP’s network. Either way, the damage caused can have severe financial and technical repercussions [12].

2.1 Problem Statement

Researchers have focused on detecting botnets in their initial stages throughout various methods, such as Deep Neural Networks (DNN) and Statistical Traffic Fingerprint analysis [13–15], seeking to stop botnet propagation and avoid severe damage caused by attacks. However, the appearance of new botnet families or substantial changes in behavioral features of existing botnets can make already trained detection models obsolete. Furthermore, collecting and preparing data of new botnets, and training new models can take a considerable amount of time. This has set a challenge to the cybersecurity community, which needs to continuously adapt their defense mechanisms and techniques, e.g. by creating new patches and updating threat signatures, to keep up with the rapidly

evolving botnets. Notably, two aspects constitute a fundamental challenge in stopping botnet propagation: automation and adaptability. On the one hand, automation is necessary to match the pace of botnets with high propagation and infection rates, and it also plays a crucial role in reducing costs implied by the mitigation [7]. On the other hand, adaptability is considered fundamental when facing the rapidly changing dynamics of botnets. Continuous adaptability is challenging to implement since it requires learning new botnet behaviors in the long run [16].

In conclusion, to adapt to modern botnets’ dynamics and propagation, ISPs need to develop mitigation mechanisms capable of **acting autonomously** and **adapting continuously** to changing botnet behaviors. **Autonomy** means that actions, such as quarantining infected machines, can be executed without human assistance, reducing bots’ probability of propagation and thus infection. **Adaptability** refers to the capacity of identifying new botnets that start propagating without the need to manually update databases, or other forms of records.

2.2 Use Case Description

Our experiment is performed in a star topology network, with Linux hosts interconnected through a central router. We observe the traffic traversing the router in an attempt to map potential botnet behavior to an action that would mitigate that traffic. The desired network state is given by the absence of threats in the traffic, and it is determined by an expected traffic volume range. A traffic volume above the upper-bound limit indicates the likely presence of botnet traffic, and a traffic volume below the lower-bound limit indicates that we are filtering not only botnet traffic but also legitimate traffic. We generate Mirai-like scanning traffic from one of the devices, which will create a constant scanning flow towards the other devices connected to the router. We chose the Mirai botnet because it has had a significant impact on the Internet, since it is considered the first IoT botnet to be a high-profile DDoS threat [12]. Furthermore, due to its code being released on the Internet, developers still use it as a reference for creating new botnets. The traffic dynamics in this use case are intended to emulate those in an ISP router, in the presence of a botnet scanning traffic. Filtering out botnet traffic in a single router allows to see the impact on legitimate traffic locally, before scaling up the filter application to multiple routers simultaneously. As our work focuses on behavior-action mapping to mitigate botnet propagation, botnet monitoring and identification are out of the scope of this paper and are assumed to be performed by a third party.

3 State of the art

The security community has been studying botnets and ways to disrupt them as early as 2005 [17]. In recent years, works on this subject have seen significant improvement in botnet detection and mitigation.

Some works use ML methods and DPI. For instance, Roosmalen et al. [14] use DNN and ladder networks for botnet detection, using raw header information from packets, differently from most academic proposals, which rely mostly on NetFlow statistics and apply feature engineering and feature selection.

Other works rely on behavior-based detection, e.g., Zhang et al. [15] explain that it is possible to detect stealthy P2P botnets even when malicious activities are not observable, (e.g., bot spamming through webmail services). They first identify all hosts that might be engaged in P2P communications in a network, then they obtain statistical fingerprints to profile different types of P2P traffic, and finally, they leverage these fingerprints to distinguish between legitimate traffic and P2P botnet traffic. Following the same behavioral principle, Haghighat et al. [13] propose a smart window technique for anomaly detection using deep learning. Their technique aggregates Netflow records and uses a sliding window for the feature extraction procedure, ensuring that this will provide the ability to learn features and detect anomalies more accurately.

Finally, some works have taken more proactive methods by using reinforcement learning (RL). Venkatesan et al. [18] propose to deploy a limited number of defense mechanisms (honeypots and network-based detectors) in the target network to reduce the lifetime of stealthy botnets by maximizing the detection rate, using RL to find the optimal deployment of these defense mechanisms. In another work, Alauthman et al. [19] propose a system capable of detecting botnets and adapt itself to their changes. They do this using a sliding time window for botnet traffic representation and a reinforcement learning algorithm for botnet detection.

In the same general path of these works but different from them, given that we focus our efforts on botnet mitigation rather than detection, we investigate the capabilities of RL for mapping botnet behavior to actions, and control the reactions that will mitigate botnet traffic, taking always into consideration automation and adaptability.

4 Experiment and Results

4.1 Experiment Description

The experiments are carried out in a network environment emulated using Vagrant and VirtualBox, with a Juniper router as the routing device and Linux machines (see Figure 1b). To interact with this environment we propose a modular, event-driven network automation (EDNA) [20] (see Figure 1a) solution to mitigate botnet propagation in the network, by automatically reconfiguring traffic filters in routing devices. The control of the automated actions is guided by a Deep Q-Learning algorithm (DQN). We chose DQN over other algorithms because of its off-policy nature, which optimizes the RL agent performance rather than optimizing a given policy. Another reason is that Q-Learning methods are more sample efficient, because they can reuse data from previous interactions with the environment to train the neural network.

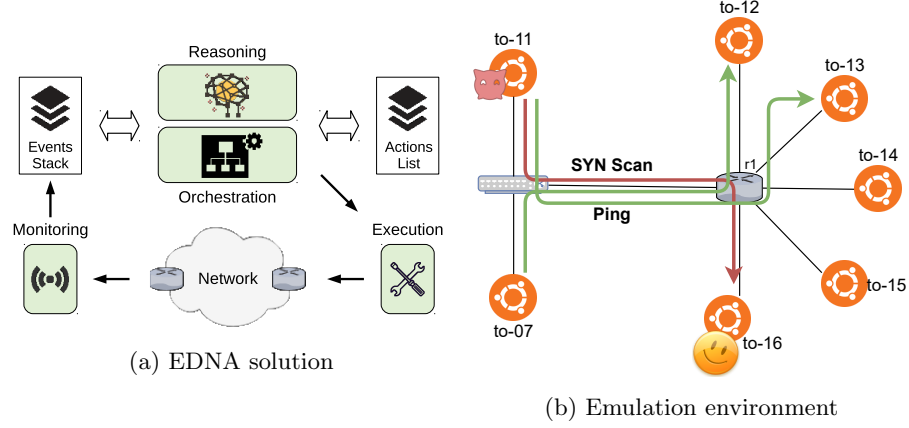


Fig. 1: (a) EDNA solution shown its building blocks and (b) emulated network environment with bot in to-11 and vulnerable device in to-16.

We emulate Mirai-like traffic with a botnet traffic generator we developed, called Botenv¹. This traffic generator uses Mirai’s fundamental behavioral features, extracted from Mirai’s source code analysis [21–23], to easily deploy and connect the main components of a botnet (C&C, bots, loader server) in a virtualized environment, allowing to build a multitude of scenarios.

The generated Mirai-like traffic will coexist with legitimate IoT traffic at the interfaces of the routing device, during the entire emulation period. The legitimate traffic has the form of continuous TCP requests and responses between socket clients and servers. The traffic volume V that traverses the interface of $r1$ in Figure 1b will be within a given range ($R = [V_{min}, V_{max}]$) when there is only legitimate IoT traffic (green arrow), and will be outside this range when legitimate traffic is aggregated with botnet traffic (red arrow, $V > V_{max}$) or when legitimate IoT traffic has been filtered ($V < V_{min}$).

EDNA reconfigures $r1$ and enables filters to cast out botnet traffic according to automatically generated policies. These filters are applied to the interface where botnet and legitimate IoT traffic coexist.

At the moment, we consider the following 3 actions identified by the following IDs (to filter by): (0) destination port, (1) source IP address, and (2) source IP prefix. The method for choosing filter parameters is out of the scope of this article, and it will be addressed in future work.

An action is deemed to be beneficial if it brings the interface’s traffic volume V within the expected range R . To reinforce this behavior in our DQN agent, we implement a reward function as defined in Algorithm 1, where r_ϵ stands for the episode’s reward. Once the action that keeps traffic volume within the expected range is found, the agent favors continuing using the same action, which means to maintain active the filter that has already been applied in the router.

¹ Botenv project on GitHub: <https://github.com/Orange-OpenSource/botenv>.

Algorithm 1 Reward function

```

if  $V_{min} < V < V_{max}$  then
   $r_\epsilon \leftarrow 1$ 
  if  $action_t = action_{t-1}$  then
     $r_\epsilon \leftarrow 2$ 
else
   $r_\epsilon \leftarrow -1$ 
end if

```

4.2 Initial Results

We present the initial results of our experiments in Figure 2. These results show the action and reward paths from episode 1 to episode 299. Each episode is constituted of 30 steps.

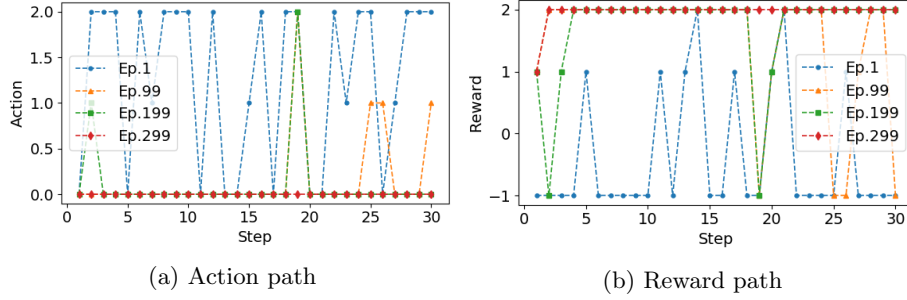


Fig. 2: Action and reward paths from episode 1 to episode 299. Each episode is constituted of 30 steps.

The action path (see Figure 2a) shows how our DQN agent explores the state-action space, initially choosing random actions, changing the traffic volume in the interface, and deeming one action better than the others based on the reward obtained. As the number of episodes increases, the influence of the learning process can be seen, since the algorithm tends to be less exploratory and ends up deciding to use a single action over the others. This action is **action 0** (*filter traffic by destination port*), which is the action that effectively filters out botnet traffic without affecting legitimate IoT traffic.

The reward path (see Figure 2b) shows a similar behavior, with values that vary during early episodes of the emulation. Here again, we can see that the DQN agent converges to the highest reward value in the late episodes of the emulation.

5 Conclusion and Future Work

Automating actions to mitigate botnet traffic in a network appears feasible. Furthermore, the use of Deep Reinforcement Learning as a control mechanism for this process shows beneficial effects.

We have evaluated the capability of a DQN agent to optimize the application of routing filters to mitigate botnets. More precisely, we have (i) proposed a generic network automation architecture that automatically handles events that put at risk its proper operation, (ii) defined a DQN agent which controls the choice of actions to be executed, and (iii) evaluated its performance through its action and reward paths. Results have proven the feasibility of such an agent and demonstrated its ability to choose the right action over time.

Future work will focus on the utilization of more robust ways to represent the state of the network traffic, e.g., using communication flows. We also want to increase the complexity of our use case by using several botnet traffic patterns, allowing us to test the capability of the DQN agent to generalize and to adapt to the botnet dynamics.

References

1. Spamhaus: Botnet threat report 2019 (2019), <https://www.spamhaustech.com/custom-content/uploads/2020/04/2019-Botnet-Threat-Report-2019-LR.pdf>
2. Cisco: 2020 global networking trends report (2020), https://www.cisco.com/c/m/en_us/solutions/enterprise-networks/networking-report.html
3. Eslahi, M., Salleh, R., Anuar, N.B.: Bots and botnets: An overview of characteristics, detection and challenges. In: 2012 IEEE International Conference on Control System, Computing and Engineering. pp. 349–354. IEEE (2012)
4. Rodríguez-Gómez, R.A., Maciá-Fernández, G., García-Teodoro, P.: Survey and taxonomy of botnet research through life-cycle. *ACM Computing Surveys (CSUR)* **45**(4), 1–33 (2013)
5. Khosroshahy, M., Ali, M.K.M., Qiu, D.: The sic botnet lifecycle model: A step beyond traditional epidemiological models. *Computer Networks* **57**(2), 404–421 (2013)
6. Khattak, S., Ramay, N.R., Khan, K.R., Syed, A.A., Khayam, S.A.: A taxonomy of botnet behavior, detection, and defense. *IEEE communications surveys & tutorials* **16**(2), 898–924 (2013)
7. Asghari, H., van Eeten, M.J., Bauer, J.M.: Economics of fighting botnets: Lessons from a decade of mitigation. *IEEE Security & Privacy* **13**(5), 16–23 (2015)
8. Osterweil, E., Stavrou, A., Zhang, L.: 20 years of ddos: a call to action. *arXiv preprint arXiv:1904.02739* (2019)
9. Huang, D.Y., Dharmdasani, H., Meiklejohn, S., Dave, V., Grier, C., McCoy, D., Savage, S., Weaver, N., Snoeren, A.C., Levchenko, K.: Botcoin: Monetizing stolen cycles. In: NDSS. Citeseer (2014)
10. Amini, P., Araghizadeh, M.A., Azmi, R.: A survey on botnet: Classification, detection and defense. In: 2015 International Electronics Symposium (IES). pp. 233–238. IEEE (2015)

11. Dhayal, H., Kumar, J.: Botnet and p2p botnet detection strategies: A review. In: 2018 International Conference on Communication and Signal Processing (ICCSP). pp. 1077–1082. IEEE (2018)
12. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., et al.: Understanding the mirai botnet. In: 26th {USENIX} security symposium ({USENIX} Security 17). pp. 1093–1110 (2017)
13. Haghighat, M.H., Foroushani, Z.A., Li, J.: Sawant: Smart window based anomaly detection using netflow traffic. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT). pp. 1396–1402. IEEE (2019)
14. van Roosmalen, J., Vranken, H., van Eekelen, M.: Applying deep learning on packet flows for botnet detection. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing. pp. 1629–1636 (2018)
15. Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., Luo, X.: Detecting stealthy p2p botnets using statistical traffic fingerprints. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN). pp. 121–132. IEEE (2011)
16. Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From throw-away traffic to bots: detecting the rise of dga-based malware. In: Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12). pp. 491–506 (2012)
17. Cooke, E., Jahanian, F., McPherson, D.: The zombie roundup: Understanding, detecting, and disrupting botnets. *SRUTI* **5**, 6–6 (2005)
18. Venkatesan, S., Albanese, M., Shah, A., Ganesan, R., Jajodia, S.: Detecting stealthy botnets in a resource-constrained environment using reinforcement learning. In: Proceedings of the 2017 Workshop on Moving Target Defense. pp. 75–85 (2017)
19. Alauthman, M., Aslam, N., Al-Kasassbeh, M., Khan, S., Al-Qerem, A., Choo, K.K.R.: An efficient reinforcement learning-based botnet detection approach. *Journal of Network and Computer Applications* **150**, 102479 (2020)
20. Aguas, E., Lambert, A., Blanc, G., Debar, H.: Automated saturation mitigation controlled by deep reinforcement learning. In: 2020 IEEE 28th International Conference on Network Protocols (ICNP). pp. 1–6. IEEE (2020)
21. Kambourakis, G., Kolias, C., Stavrou, A.: The mirai botnet and the iot zombie armies. In: MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM). pp. 267–272. IEEE (2017)
22. Marzano, A., Alexander, D., Fonseca, O., Fazzion, E., Hoepers, C., Steding-Jessen, K., Chaves, M.H., Cunha, Í., Guedes, D., Meira, W.: The evolution of bashlite and mirai iot botnets. In: 2018 IEEE Symposium on Computers and Communications (ISCC). pp. 00813–00818. IEEE (2018)
23. Bastos, G., Marzano, A., Fonseca, O., Fazzion, E., Hoepers, C., Steding-Jessen, K., HPC, C.M., Cunha, Í., Guedes, D., Meira, W.: Identifying and characterizing bashlite and mirai c&c servers. In: 2019 IEEE Symposium on Computers and Communications (ISCC). pp. 1–6. IEEE (2019)

Data Exchange for Anomaly Detection: Analysis of CAN bus logs

Yannick Chevalier^[0000–0002–8617–4209]

IRIT, Université Toulouse 3, Toulouse, France
`yannick.chevalier@irit.fr`

Abstract. The Controller Area Network (CAN) bus is efficient but insecure. Its efficiency has made it the network of choice for the automotive industry, and its insecurity has triggered the development of a variety of Intrusion Detection Systems (IDS) to analyze the traffic and detect anomalies. In existing implementations, obfuscation is the main source of security: though most messages have a well-defined meaning, the association between messages and their meaning (the CAN matrix) is usually secret. We propose in this paper a framework based on Data eXchange (DX) to build a symbolic model from simple tests. Analysis of the first results obtained allowed us to integrate user expertise to reconstruct automatically messages of the Multi-Frame Message (MFM) protocol.

Keywords: CAN bus · Machine Learning · Intrusion Detection · vehicular security

1 Introduction

1.1 Context and presentation of the results

The Controller-Area Network (CAN) bus protocol is a bus protocol invented in 1986 by Robert Bosch GmbH. Originally intended for automotive use, it has since been adopted for internal communication of various Cyberphysical systems, ranging from sewer station to *e*-bikes. Nowadays it has been adopted by all automotive manufacturers and suppliers, on all models of cars and trucks.

The designers of the CAN bus focused on reliability to ensure the timely dissemination of information within a vehicle isolated from its environment. Security considerations were limited to the protection of “secret of the trade” by not publishing the role of each message in the dissemination of information within the vehicle, *i.e.* the CAN matrix relating CAN and OBD-II message IDs.

Approach to anomaly detection. It is advocated in [7] that in order to be effective, network IDS need to *understand the system*. It is also advocated that they need to have low false positive (FP) and false negative (FN) rates: in the first case to reduce the cost of treating the errors, and the second because any anomaly can be the sign of a compromise within the network. We thus aim at building an IDS

whose model of the normal activity on the CAN bus can be assessed by humans, and describe why some messages were considered anomalies so that meaningful mitigation can be enacted. This need for explainability lead us to deviate from usual machine learning activities.

Starting from *characteristic functions* [2], we have developped a novel approach for log analysis based on *symbolic machine learning*. A log analyzer builds a logic theory of the CAN bus messages based on normal messages, and a monitor accepts only messages that satisfy that theory. We structure the theory built by the analyzer as a *Data Exchange System* (DXS). The *Data Exchange* problem consists in computing, given two databases, a source and a target, with each its own schema, and rules expressing how to fill tables on the target given the records in the source. A DXS is simply the result of iterated instances of the DX problem, with each table in the target database added to the source database after the exchange. Generating rules express possible transformations on the data, while constraint rules express tests and integrity constraints that must be true on the transformed data.

For anomaly detection, a *source table* in a DXS contains the events recorded on the network. The other tables are filled as events are recorded, and constraint rules are checked. The event is rejected as an anomaly if one of these constraints is not satisfied, and accepted otherwise. This DXS is built during a learning phase by building a filter that accepts all recorded events. The analyzer and the monitor run in time *linear in the size of the log*, and each record is visited only once, which allows for online learning and monitoring.

Experimental results. The *Oak Ridge National Laboratory* has recently submitted [9] a set of automotive logs including ambient logs and logs containing real injection attacks on a vehicle. Applying the learning phase on all attack logs, we were able to capture all attacks but for those that were non-local, *i.e.* in which the attack is the activation of the turn left signal when the driver signals a right turn, and the other way around. The detection of these attacks would require a better understanding on the relationships among CAN bus events that isn't implemented yet.

Outline. We discuss related works in the rest of this section. In Sec. 2 we present how we employ a DXS to monitor a CAN bus. In Sec. ?? we briefly present how to synthesize a DXS for a CAN bus from logs of this CAN bus, and we present our experimental results based on this approach in Sec. 3. We prove properties that practitioners may find useful in Sec. ?? before concluding in Sec. 4.

2 Data Exchange Systems

2.1 Basic Definitions

We introduce Data exchange systems (DXS) to model the expected traffic on a CAN bus. We reason on the domain of the integers, together with the functions $+$, $-$, \dots , predicates $<$, $=$, \dots and their usual interpretation. Integers in the

domain are denoted using a_1, \dots or with their actual value. Ground terms are expressions denoting the application of some functions on integers and are denoted t, \dots , and atoms relate different terms with a predicate and are denoted A, B, \dots . We also introduce a set of *table predicates* usually denoted R and decorations thereof, each of fixed arity, and a special table predicate S that is minimal for an ordering \prec_T on the set of tables. A *model* ρ is a finite set of *ground table atoms* $R(t_1, \dots, t_n)$ where R is a table predicate of arity n and t_1, \dots, t_n are ground terms. A model ρ satisfies a ground table atom A if $A \in \rho$ and we denote it $\rho \models A$. Otherwise, if $A \notin \rho$, we write $\rho \models \neg A$. The truth of non table predicates is evaluated as that of their integer interpretation. The truth value of ground formulas is defined as usual.

We also assume an infinite set of variables denoted x, y, n, \dots . Terms are likewise defined as expressions over integers and variables, and (table) atoms relate terms with variables. Ground substitutions are mappings from variables to ground terms, usually denoted σ, \dots . The application of substitution on terms and atoms is denoted in the postfix notation, *i.e.* $t\sigma$ rather than $\sigma(t)$. An atom A (*resp.* a formula φ) is true in the model ρ , and we denote $\rho \models A$ (*resp.* $\rho \models \varphi$), whenever for all ground substitution σ we have $\rho \models A\sigma$ (*resp.* $\rho \models \varphi\sigma$). We denote tuples (of variables, integers, etc) with a bold face, *i.e.* \mathbf{x} instead of x_1, \dots, x_n when the exact arity is of little importance.

2.2 Data Exchange Systems

We now simplify the data exchange setting presented *e.g.* in [5] to denote the two specific types of formulas that we consider.

Generating rules. A *table observation* is a non-empty conjunction of table atoms. It is unitary if it contains only one atom. A *value observation* is a positive formula that does not contain any table atom. A *generating rule* for a table predicate R is a rule of the form:

$$\forall \mathbf{x}, (\exists \mathbf{y}, \varphi(\mathbf{x}, \mathbf{y})) \Rightarrow \psi(\mathbf{x})$$

where φ is a conjunction of a table observation and of a value observation, and $\psi = R(\mathbf{x})$ is a unitary table observation. We only consider generating rules such that the table predicates in the antecedent of the implication are strictly smaller for the order \prec_T than the one in its conclusion.

Constraint rules. A *constraint rule* is a rule of the form $\forall \mathbf{x}, (\exists \mathbf{y}, \varphi(\mathbf{x}, \mathbf{y})) \Rightarrow \psi(\mathbf{x})$ where φ is a conjunction of a table observation and of a value observation, and ψ is a value observation.

Example 1. Given a table predicate R one can “create” a table predicate R_d holding the differences of the fields in R :

$$\forall x_{id}, \mathbf{x}, \mathbf{y}, (R(x_{id}, \mathbf{x}) \wedge R(x_{id} + 1, \mathbf{y})) \Rightarrow R_d(x_{id}, \mathbf{y} - \mathbf{x})$$

Similarly, we say that the field k in table R is a *counter modulo N* if the difference between records in that table is a constant modulo N , *i.e.* if the following constraint rule is always satisfied:

$$\forall x_{id}, y_{id}, \mathbf{x}, \mathbf{y}, (R_d(x_{id}, \mathbf{x}) \wedge R_d(y_{id}, \mathbf{y})) \Rightarrow (x_k \bmod N) = (y_k \bmod N)$$

We leave to the reader the constraints implying that a value in a field is bounded, and that its derivative is also bounded. A last useful rule is the *splitting* rule, that create a new predicate for each value of a field, and is employed implicitly to build one table for each CAN bus message ID.

We note here that these rules have been simplified to clarify their presentation. Our restrictions cannot express *e.g.* that a record is the last in the table. However sequences of records can be encoded in additional fields and table.

Definition 1. (*Data Exchange System*) A Data Exchange System \mathcal{D} is a tuple $(\mathcal{T}, \mathcal{L}_g, \mathcal{L}_c)$ where \mathcal{T} is a set of tables, \mathcal{L}_g is a set of generating rules, and \mathcal{L}_c is a set of constraint rules.

We assume that the source table S as well as the table predicates in each rule are in \mathcal{T} , and conversely that \mathcal{T} contains only S and the table predicates occurring in the conclusion of a generating rule in \mathcal{L}_g .

2.3 DXS for intrusion detection

Let $\mathcal{D} = (\mathcal{T}, \mathcal{L}_g, \mathcal{L}_c)$ be a DXS. We use \mathcal{D} as an intrusion detection system as follows:

- every message appearing on the CAN bus is added to the table S ;
- the tables in \mathcal{T} are filled using the generating rules in \mathcal{L}_g and the *chase procedure* [1,6];
- the message is accepted if all constraint rules in \mathcal{L}_c are satisfied, and rejected as an *anomaly* otherwise.

In a white box setting, with detailed information on the CAN bus, it is easy to construct a DXS that monitors the bus and detects non-conforming messages. Once these messages are detected, using *e.g.* voltage fingerprinting, it is possible to disable the ECU having sent these messages.

Our main contribution is to construct in a black box setting a DXS that can accurately flag non-conforming messages even in attacks that were classified as *transparent* by their author [9], and carefully crafted by a team of experts over several months.

2.4 Intuition on learning a DXS by Log Analysis

To show our approach on the characterisation of the different fields in a CAN bus message, we consider a sequence of values for that field that can be seen

on a log. To begin with, consider the sequence 0,0,0 of values. The simplest and strictest interpretation of this sequence is that this field contains the constant 0. Let us extend it to 0,0,0,1,1,1,1,2,2,2,2, or actually its derivative 0,0,1,0,0,0,1,0,0,0,1. The simplest explanation now is that this sequence corresponds to a counter that increases by 1 every four ticks. Note here that a possible explanation for the first sequence was already being a counter, but with a change every more than 3 ticks. If we continue to extend this sequence *e.g.* with 0,0,0,1,1,1,1,2,2,2,2,0, we would conclude this is probably a physical value, that has increased a bit, then decreased. The key here being that the set of values visited in the range is dense, and that the change is also dense. Finally, if we extend the sequence to 0,0,0,1,1,1,1,2,2,2,2,0,4,68, the last possible explanation is that, from one value to the next, only a few bits can change.

We approach these tentative explanations as follows. From the start, and before seeing the first value, we assume the sequence is at the same time constant **and** equal to 0, constant **and** equal to 1, ..., *and* a counter with a difference of k every n steps for all $k, n \in \mathbb{Z}, n \geq 1$, *and* a physical value with all possible ranges, *and* a value encoding a *state* with all maximum of bit changes at each step. All of these starting conditions are false, \perp , but actual values for the field filter out those that are not satisfied on the sequence. The first 0 eliminates all the possible constants but 0, the second 0 eliminates all counters that change at every tick, etc. This process of possibilities elimination is captured in the notion of *filter*: given a current characterisation and a counter-example, there always exist a next best characterisation, even \top if no good characterisation exists in the filter. We apply filters on messages and fields with *constraint rules*.

Beyond the individual fields, other characteristics are likely to be found in a log: if a sequence of messages corresponds to a protocol, we can expect to find regularities, such as a reused session identifier, among the messages of that sequence. Relying only on filters may be possible, but is likely to produce ever more intricate ones. Another possibility is to create new tables to partition the set of messages, and to regroup messages that have been deemed to be related. For example, a first step is to create one new table, for each message ID, as on the CAN bus a message ID denotes a message format, and it is unlikely to find a pattern to all message ID. We have also employed new tables to recreate a multi-frame message from its different parts, and analyse the sequence of multi-frame messages. These tables are introduced by *generating rules*.

3 Experimental results

In order to work with messages of variable lengths when transmitted using the *Multi-Frame Message* protocol, we have chosen to use GNU awk to implement our analyser and our monitor. Currently, a log is read twice: once for the detection and synthesis of MFM messages, and a second one for the proper analysis. New tables are created by altering the ID field of messages, and they are at the moment explicitly constructed only for MFM message, with an ID varying

according to the length of the multi-frame message, or the place in the protocol. We are currently working on a C implementation that would be able to read the log only once, and performs all computations online.

We currently only consider bytes boundary for values analysed, whereas it is well known that values are often transported across these boundaries [8]. Also in comparison with [8], we consider two possible interpretations of bytes as either signed or unsigned, but only consider little-endianness. We see no difficulties in extending our approach to also consider these cases, and have already worked in a previous implementation on the detection of byte boundaries with colleagues [3], and plan to re-integrate it in future works.

The current implementation of our tool is around 2000 lines of codes, mostly in `awk` but also with some bash scripting to glue different scripts together. Each filter is implemented in a distinct file, where both the analysis and the monitoring part are defined in separate functions. Two `awk` scripts include these files and call these functions for respectively building a model and monitoring a log. The learner reads the log and outputs a specification in the csv format. That file is read by the and specifies the checks to perform on the messages.

We have analysed all the attack logs in the ORNL dataset [9] to produce a monitor specification, and then reused that specification for each of the attack logs. The results and timings are in Table 1. It is interesting to note that in most attacks, all (but for 3 in total) messages are labeled correctly, only the reverse light signal attack is completely invisible to our tool. When detecting a real anomaly, the cause was:

- a field had a value out of its admitted bounds in 32024 cases;
- a field had too much difference with its value in prior messages in 2449 cases;
- a half-byte recognition anomaly in 60132 cases;
- a checksum error in 18570 cases;
- an unregistered id in 19022 cases;

Counters do not appear in the final version of the analyzer because they are tested last, but when monitoring only with counters the anomalies on message with ID 208 are discovered, as in the attack the counter value is reused from the previous message instead of being incremented by 2. This detection captures all attacks that would aim at changing a value in an ECU as they need to reuse the counter to change the current value.

Other analyses performed. In addition to the ORNL logs, we have also analyzed HCRL [4] logs with only true results, and a Renault Zoé log captured and provided by colleagues at Fraunhofer Darmstadt SIT team. Out of 10^6 records there were only 9 false negative, all but two of them on message ID 700, for which no test were found using our method. One of the other is a two bytes messages, and though one anomaly has been accepted, the bounds seem realistic if interpreted as those of a 2-bytes physical (signed or unsigned, it stays positive) value. The remaining false negative is a 5-bytes message whose 4 last bytes again appear to be random. We were also able to detect an implementation of

a variation of the MFM protocol. This lead us to implement a string recogniser that tracks fields where the value is always a printable ASCII character or 0. We were able to obtain meaningful data from the logs, most of which was related to RDS data, but some seem related to a mobile phone identification number.

Table 1. Attack logs are learned twice, time in seconds. LT: learning time, MT: monitoring time, FP/FN: false positive/negative, RP: real positive.

log	FP	FN	RP	Total	LT	LT	MT
accelerator attack drive 1	0	0	0	204760	25.57	28.74	13.84
accelerator attack drive 2	0	0	0	171936	21.96	22.61	11.76
accelerator attack reverse 1	0	0	0	202447	25.47	26.72	13.66
correlated signal attack 1	0	0	2087	81262	10.07	10.36	5.30
correlated signal attack 2	0	0	2141	69657	8.59	8.81	4.55
correlated signal attack 3	0	0	1265	41829	5.14	5.30	2.75
fuzzing attack 1	0	2	592	49342	6.06	6.14	3.31
fuzzing attack 2	0	1	352	32351	3.99	4.15	2.14
fuzzing attack 3	0	0	115	13238	1.85	1.79	0.89
max engine coolant temp attack	0	0	43	61923	7.81	7.85	4.14
max speedometer attack 1	0	0	2445	213397	27.12	26.55	14.25
max speedometer attack 2	0	0	3141	145894	18.42	18.39	9.77
max speedometer attack 3	0	0	6108	213551	26.69	26.64	14.33
reverse light off attack 1	0	673	673	67902	8.43	8.60	4.57
reverse light off attack 2	0	2372	2372	99628	12.39	12.17	6.84
reverse light off attack 3	0	2435	2435	140855	17.18	17.68	9.50
reverse light on attack 1	0	1992	1992	133237	16.60	16.48	9.08
reverse light on attack 2	0	3690	3690	175945	22.26	22.21	11.99
reverse light on attack 3	0	2352	2352	156052	20.37	19.78	10.44

4 Conclusion and Future Works

One of our main experimental findings is that with current machine learning techniques, there is a misplaced trust on security by obscurity as it is now possible to reveal details on the CAN bus messages format and their payload. For example, this is the first tool we are aware of that is able to reconstruct messages of the MFM protocol, and get additional insight on its implementation as well as on the payload of these messages.

Knowledge and future works. The lack of knowledge of a sequence of values has long been formally defined in cryptography by the inability of an observer to produce a test distinguishing that sequence from a random one. Our approach can be presented as a contrapositive: knowledge is defined by the production of meaningful tests on sequences of messages. Pursuing this analogy, we plan to

consider in future works *adaptive machine learning* in which we allow interaction with the CAN bus during the analysis part to gain more information on the system. An intermediate result will be to generate CAN bus data from the model learned, and to *simulate* the actual bus traffic with this model. This will also allow us to detect missing messages as messages expected by the simulator and not occurring on the actual bus. Finally we believe that the precision achieved by our tools will enable us to conduct penetration testing on real vehicles.

References

1. Aho, A.V., Beeri, C., Ullman, J.D.: The theory of joins in relational databases. *ACM Trans. Database Syst.* **4**(3), 297–314 (1979). <https://doi.org/10.1145/320083.320091>, <https://doi.org/10.1145/320083.320091>
2. Chevalier, Y., Rieke, R., Fenzl, F., Chechulin, A., Kotenko, I.V.: Ecu-secure: Characteristic functions for in-vehicle intrusion detection. In: Kotenko, I.V., Badica, C., Desnitsky, V., Baz, D.E., Ivanovic, M. (eds.) *Intelligent Distributed Computing XIII, 13th International Symposium on Intelligent Distributed Computing, IDC 2019, St. Petersburg, Russia, 7-9 October, 2019. Studies in Computational Intelligence*, vol. 868, pp. 495–504. Springer (2019). https://doi.org/10.1007/978-3-030-32258-8_58, https://doi.org/10.1007/978-3-030-32258-8_58
3. Fenzl, F., Rieke, R., Chevalier, Y., Dominik, A., Kotenko, I.V.: Continuous fields: Enhanced in-vehicle anomaly detection using machine learning models. *Simul. Model. Pract. Theory* **105**, 102143 (2020). <https://doi.org/10.1016/j.simpat.2020.102143>, <https://doi.org/10.1016/j.simpat.2020.102143>
4. Hacking and Countermeasure Research Lab (HCRL): Car-hacking dataset for the intrusion detection. <http://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset> (2018), [Online; accessed 28-Jun-2018]
5. Kolaitis, P.G., Panttaja, J., Tan, W.C.: The complexity of data exchange. In: Vansummeren, S. (ed.) *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, June 26–28, 2006, Chicago, Illinois, USA. pp. 30–39. ACM (2006). <https://doi.org/10.1145/1142351.1142357>, <https://doi.org/10.1145/1142351.1142357>
6. Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing implications of data dependencies. *ACM Trans. Database Syst.* **4**(4), 455–469 (1979). <https://doi.org/10.1145/320107.320115>, <https://doi.org/10.1145/320107.320115>
7. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: *2010 IEEE Symposium on Security and Privacy*. pp. 305–316 (2010). <https://doi.org/10.1109/SP.2010.25>
8. Verma, M.E., Bridges, R.A., Sosnowski, J.J., Hollifield, S.C., Iannacone, M.D.: CAN-D: A modular four-step pipeline for comprehensively decoding controller area network data. *CoRR* **abs/2006.05993** (2020), <https://arxiv.org/abs/2006.05993>
9. Verma, M.E., Iannacone, M.D., Bridges, R.A., Hollifield, S.C., Kay, B., Combs, F.L.: ROAD: the real ORNL automotive dynamometer controller area network intrusion detection dataset (with a comprehensive CAN IDS dataset survey & guide). *CoRR* **abs/2012.14600** (2020), <https://arxiv.org/abs/2012.14600>

Real-time graph clustering for network intrusion detection

Amine Medad, Baptiste Gregorutti, Edouard Genetay, and Alexandre Peter Nguema

LumenAI

25 rue de la Milletiere, l'Aeronef, Bat. Auriol, 37100 Tours
bgregorutti@lumenai.fr

Abstract. In this paper, we propose an online graph clustering approach for detecting suspicious activities in a computing system based on the detection of communities in a network. Assuming that attacks dynamically create abnormal connections between system processes within the network, the detection of the communities will detect intrusions. We apply this approach on a publicly available dataset OpTC provided by the Defense Advanced Research Projects Agency (DARPA). Our preliminary results show the feasibility of our approach and encourages the exploration towards an intelligent attack detection end-to-end system.

Keywords: Real-time · Graph clustering · Intrusion detection.

1 Introduction

Anomaly detection is the identification of rare events or failures in a system. This is highly challenging especially when processing a large amount of data. According to the survey [1], various domains are considered such as health insurance [16], security [9], etc. More particularly, one can mention intrusion detection as a sub-domain of anomaly detection applied to cybersecurity. The typology of data is numerous: DNS logs¹, authentication logs [15], system processes logs [23], network traffic logs [22], etc. The volume of data ranges from 1Mo to several To for only a tiny fraction of anomalies [2].

Due to the network structure of the data, a graph modelling is well suited to detect anomalies. Indeed anomalies can be defined as abnormal connections between normal and suspicious objects. The graph itself carries lot of correlation information from these connections, see [1]. For instance, when considering a graph of system processes, which is the purpose of our work, the nodes of the graph are processes and the edges are the actions of processes to another one (create, open or terminate).

¹ as provided by <https://www.shodan.io/>

Community detection is one of the methods used in graph machine learning. It consists of partitioning the graph into a set of nodes that share a large number of connections between them. Rossetti [20] proposes a survey on dynamic community detection algorithms including modularity-based algorithms, see [13, 4, 21, 12, 19, 3]. Other studies focused on the analysis of large social networks or research articles citations, for instance [7, 11], and more recently the analysis of emails [10]. More methods and codes are publicly available². Here we propose to detect intrusions in a computing system using a dynamic MCMC algorithm introduced in [8] which is a dynamic modified version of the Louvain partitioning procedure from [5]. This is the first use of this algorithm on real data.

The paper is organised as follows. Section 2 presents the motivations of our work i.e. applying a graph clustering algorithm to system processes. Section 3 introduces the mathematical concepts behind the online clustering-based community detection. Section 4 describes the Operationally Transparent Cyber (OpTC) dataset used in our experiments. Section 5 is twofold. Firstly, we describe how to build a proper graph from the OpTC data. Secondly, the results of the proposed methodological approach are detailed to detect suspicious events.

2 Motivations and goals

As presented in [1, 17], detecting intrusion in a computing system should take advantage of the network structure of the data. The definition of a graph is straightforward: the nodes are the components of the system (servers, files, processes, etc.) and the edges are the connections between the components.

Our goal is to analyse processes in a system. In this case, the graph is defined as the actions from processes to another one: *create* or *terminate* when a process creates or terminates another process, *open* when a process accesses the memory space of another process. In [17], the authors claim that a non-supervised learning approach should be preferred than a supervised approach since it does not require a labelled dataset. Similarly, we propose a clustering approach to find groups of nodes that are strongly connected together. This is commonly known as “graph clustering” or “community detection”. This is motivated by the fact that intrusions change the structure of the graph by creating new clusters (or communities in the sequel). In particular, processes due to an abnormal activity, create or terminate “normal” processes. Consequently, new communities are created due to new connections between nodes. On the other hand, we take into consideration that the graph changes in time, i.e. new nodes and edges are dynamically collected.

We propose to use the algorithm [8] for this purpose which is a real-time MCMC Louvain algorithm in order to dynamically detect communities within a graph representation of a computer network.

² <https://github.com/1172939260/community-detection>

3 Online graph community detection

Graph and community. Let $G = (V, E)$ be a weighted undirected graph where V is the set of n nodes and E is the set of edges corresponding to the relationship between the nodes. Let $A \in \mathcal{M}_n(\mathbb{R})$ be the corresponding symmetric adjacency matrix where A_{ij} denotes the weight assigned to edge $(i, j) \in E$. The degree d_i of a node $i \in V$ is basically the sum of the weights of the edges incident to i .

There is no absolute definition of a community in a network. We choose to define a community as a set of nodes that are strongly connected together. Consequently, detecting communities in a graph is finding a partition of its nodes. One way to assess the quality of a partition and then to detect communities is to use the modularity function (see for instance [5, 14, 9]). Broadly speaking, for one partition of the graph into communities, the modularity measures the quality of the groups formed. Namely, if there is a high level of connectivity inside the groups and a lower level of connectivity between the groups, then the partition is considered as “good”.

More formally, let $C = \{c_1, \dots, c_K\} \in \mathcal{C}$ be a partition of nodes, i.e. a set of communities and \mathcal{C} the set of all possible partitions. The modularity function $Q : \mathcal{C} \rightarrow [-1, 1]$ is defined as:

$$Q(C) = \frac{1}{2m} \sum_{k=1}^K \left(\sum_{i,j \in c_k} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \right),$$

where $m := |E| = \frac{1}{2} \sum_i d_i$. Accordingly, when the partition C yields a high modularity, the graph will have dense connections between nodes within communities but sparse connections between nodes in different communities.

Community detection. The problem of detecting communities can be viewed as finding the partition of nodes C^* that maximises the modularity function:

$$C^* \in \arg \max_{C \in \mathcal{C}} Q(C). \quad (1)$$

This optimisation problem, also known as graph clustering, is NP-hard due to the combinatoric issues related to the set \mathcal{C} [6]. For this reason, heuristic approximations was proposed such as the well-known Louvain algorithm [5]. This greedy search algorithm iterates two steps until a convergence is achieved. The optimisation step aims at maximising the modularity function by locally moving each nodes to one of its neighbours’ clusters, see Algorithm 1. New communities are built by gathering the closest nodes together only if the modularity index increases. Secondly, the aggregation step merges each cluster to one node and builds a new graph whose nodes are the communities themselves. Then, the algorithm is applied to this new graph and builds again a new graph whose nodes are communities of communities and so on.

Algorithm 1 LOUVAIN ALGORITHM: OPTIMISATION STEP

```

1: Initialize the communities  $c_i = \{i\}$ , for all  $i \in V$ 
2: for each node  $i \in V$  do
3:   for each node  $j$  adjacent to  $i$  do
4:     Move the node  $i$  to the community of the node  $j$ 
5:     Compute the modularity change
6:   end for
7:   Assign the node  $i$  to the community with the highest modularity increase
8: end for

```

Online graph community detection with MCMC. In [8], the authors propose an online version of the Louvain algorithm. The challenge is to take into account the change of the graph in real time. In a nutshell, the deterministic local move 4 in Algorithm 1 is replaced by a probabilistic move using a Markov Chain Monte Carlo approach (MCMC). The choice of the node and the target community is done by a particular prior distribution that maintains the hierarchy of the communities. The local move is then accepted or not given the value of an acceptance ratio as it is classically done in MCMC approaches. Also, the proposed algorithm in [8] is a modification of the Metropolis-Hasting that can handle dynamic graphs whose nodes, edges and weights change in time. With such an algorithm, the “number of community” is a natural quantity to monitor and it can be done in parallel with the modularity optimisation.

4 Operationally Transparent Cyber dataset

As part of the OpTC research program, DARPA has released a dataset over the past few years [23]. The objective was to provide realistic basis for cyber defence studies. The last update of the data repository proposes a new event based model, called extended Cyber Analytics Repository (eCAR). A pool of 1000 hosts with OS Windows 10 has been monitored for 6 days. During 3 of them, attacks occurred with 3 different scenarios³.

Day 1: To get elevated privileges and lateral movements between target hosts using Powershell Empire. To get credentials using Mimikatz. 18 machines were compromised.

Day 2: To extract personal data by sending phishing emails containing a malware. Once checked in, the administrator’s credentials enable data exfiltration via Netcat and Remote Desktop Protocol.

Day 3: To access personal data including credentials. A malicious binary file is downloaded during an update of Notepad Plus. After connecting to the malicious server during the update, the host opens a breach allowing the attacker to

³ <https://github.com/FiveDirections/OpTC-data>

Objects	Actions	Total %
FLOW	message, open, start	71.7
FILE	create, delete, modify, read, rename, write	12.4
PROCESS	create, open, terminate	8.6
MODULE	load	3.9
REGISTRY	add, edit, remove	0.3
HOST	start	0.0

Table 1: Objects found in the eCAR dataset

access the system’s personal data, including credentials.

Each host of the experimental setup was equipped with a sensor that retrieves a multitude of events before putting them in eCAR format. The malicious actions executed by DARPA on this dataset clearly highlight the behaviour of an Advanced Persistent Threat (APT). Thus, the diversity of benign and malicious interactions makes this dataset very interesting and potentially useful for the detection of anomalies in computer networks.

The OpTC dataset contains 17 billions of events of network communications and system logs, including 292,387 malicious ones. Each event describes an interaction between a process and an object with specific fields depending on the nature of the object. Therefore we find elements such as object type, action, timestamp or command lines at the origin of an event. Table 1, describes the main objects found in the dataset. Each type of object is associated with specific actions that a process can perform on it. It is important to note that only 0.0016% of events are malicious. The distribution of malicious and benign events is highly imbalanced and makes predictions about the minority class difficult. But this situation puts us in a realistic context in order to set up the most relevant metrics that will allow us to raise an alert in case of malicious intrusion. Also, this motivates the use of a unsupervised approach instead of a supervised prediction approach.

5 Cybersecurity intrusion detection

Definition of a graph of Processes. For our simulation, we focus on the events involving processes on host 201 on day 1 only. Each event that occurs implies an interaction between a process and one of the objects listed in Table 1 which leads to an intuitive definition of a graph. We use the attributes “actorID” (the ID of a process) and “objectID” (the ID of an object) of each event to create an edge as shown in the video of our supplementary material⁴ and in Figure 1. Given this definition, the graph represents all the historical activity of the network at a time t .

Graph clustering and interpretation. Our procedure continuously updates communities as explained in section 3. Figure 2 represents the number of the com-

⁴ see description and video online: <https://www.youtube.com/watch?v=LPBuE0kBIr4>

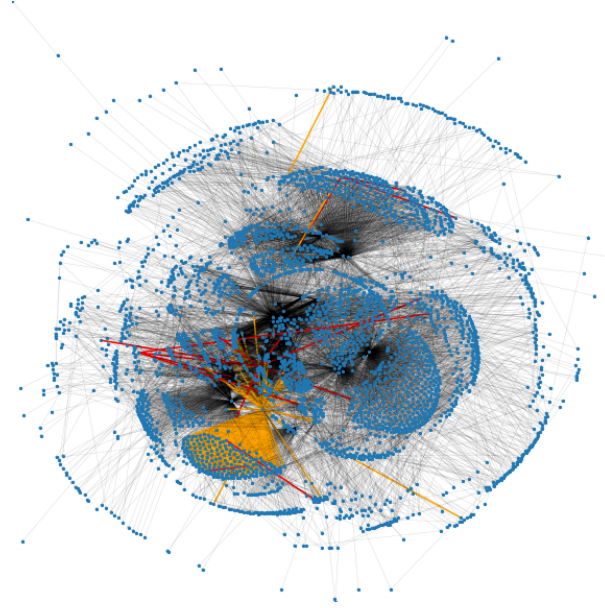


Fig. 1: Plot of the graph of processes (machine 201, day 1), processes with PID 2952 in orange and 5452 in red

munities at each edge income (top) and the histogram of the activity (bottom) of the system 201. The dots in orange in the top figure correspond to a kind of seismogram of the events with PID 5452 or 2952, i.e. processes due to the attacker. Therefore we monitor the number of community and we observe an effect consistently with our hypothesis. This was also used by [18] in the case of social networks analysis. Here, we look for discontinuities in the number of the communities. As shown in Figure 2, the number of communities has a regular behaviour and few discontinuities. The most frank discontinuities occur around 11:50am and 1:15pm. The first big one after start, at 11:50am, is due to a normal but intensive activity of the system that change a lot the graph size and topology (see the bottom chart and our video). The second big discontinuity, at 1:15pm, coincides with a high abnormal activity. The orange edges in the Figure 1 is the subgraph corresponding to PID 2952. According to our investigation, attacker scanned all the IP addresses on a sub-network leading to more or less 256 pings. Some of these pings should have given birth to their own community while others should have been attributed to an existing one. That is why the number of community metric shows a sudden increase at the time of the attack.

On day 1, host 402 and 660 were also attacked after host 201 but the actions of the attacker were not revealed by the monitoring of the number of community. We are still investigating other metrics that could reveal them.

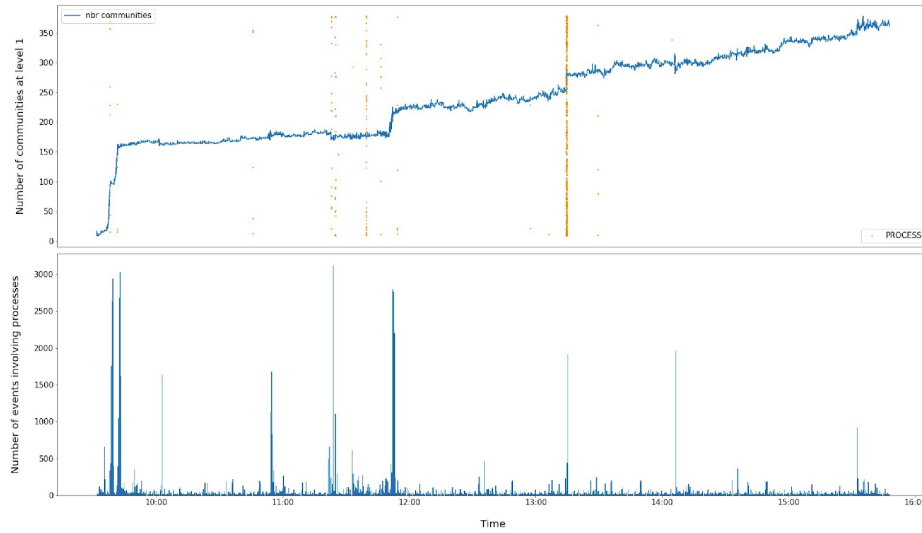


Fig. 2: Number of communities (top) and processes activity (bottom) for machine 201. The dots in orange are a seismogram of the malicious activities.

6 Conclusion

In this paper we presented a graph based approach for the real-time detection of intrusions in a computer network. According to our observations, the number of communities seems to be a good metric to detect some intrusion through the analysis of processes data. Indeed, as the attack creates new processes from existing processes, it has an effect on the graph topology by creating new communities. This is in line with our choice to use the number of communities for detecting malicious actions. As a future work, we should explore additional monitoring metrics to refine the analysis, for instance the leaders of the communities. A more extensive numerical analysis will be performed to evaluate more precisely the performances of such clustering-based algorithm.

References

1. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery* **29**, 626–688 (2015)
2. Anjum, M.M., Iqbal, S., Hamelin, B.: Analyzing the usefulness of the darpa optc dataset in cyber threat detection research. In: *Proc of the 26th ACM Symposium on Access Control Models and Technologies*. pp. 27–32 (2021)
3. Aynaud, T., Guillaume, J.L.: Multi-step community detection and hierarchical time segmentation in evolving networks. In: *Proceedings of the 5th SNA-KDD workshop*. vol. 11 (2011)
4. Bansal, S., Bhowmick, S., Paymal, P.: Fast community detection for dynamic complex networks. In: *Complex networks*, pp. 196–207. Springer (2011)
5. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10), P10008 (2008)

6. Brandes, ., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering* **20**(2), 172–188 (2008)
7. Chamberlain, B.P., Levy-Kramer, J., Humby, C., Deisenroth, M.P.: Real-time community detection in full social networks on a laptop. *PLoS One* **13**(1), 1–37 (2018)
8. Darmaillac, Y., Loustau, S.: MCMC louvain for online community detection. *CoRR* **abs/1612.01489** (2016)
9. Ding, Q., Katenka, N., Barford, P., Kolaczyk, E., Crovella, M.: Intrusion as (anti) social communication: characterization and detection. In: *Proc. of the 18th SIGKDD*. pp. 886–894 (2012)
10. Fang, G., Ward, O., Zheng, T.: Online community detection for event streams on networks. *CoRR* **abs/2009.01742** (2020)
11. G., P., W., Z., W., S., L.: Online community detection for large complex networks. *PLoS One* **9**, 1–37 (2014)
12. Gong, M., Ma, L., Zhang, Q., Jiao, L.: Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Physica A: Statistical Mechanics and its Applications* **391**(15), 4050–4060 (2012)
13. Görke, R., Maillard, P., Staudt, C., Wagner, D.: Modularity-driven clustering of dynamic graphs. In: *International symposium on experimental algorithms*. pp. 436–448. Springer (2010)
14. Hofman, J.M., Wiggins, C.H.: Bayesian approach to network modularity. *Physical Review Letters* **100**(25) (2008)
15. Kent, A.D.: *Cybersecurity Data Sources for Dynamic Network Research*. In: *Dynamic Networks in Cybersecurity*. Imperial College Press (Jun 2015)
16. Kumar, M., Ghani, R., Mei, Z.: Data mining to predict and prevent errors in health insurance claims processing. In: *Proc of the 16th SIGKDD*. pp. 65–74 (2010)
17. Leichtnam, L., Totel, E., Prigent, N., Mé, L.: Novelty detection on graph structured data to detect network intrusions. In: *Conference on Artificial Intelligence for Defense* (2020)
18. McAuley, J., Leskovec, J.: Learning to discover social circles in ego networks. In: *NIPS*. vol. 2012, pp. 548–56 (2012)
19. Nguyen, N.P., Dinh, T.N., Xuan, Y., Thai, M.T.: Adaptive algorithms for detecting community structure in dynamic social networks. In: *2011 Proceedings IEEE INFOCOM*. pp. 2282–2290. IEEE (2011)
20. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. *ACM Computing Surveys (CSUR)* **51**(2), 1–37 (2018)
21. Shang, J., Liu, L., Xie, F., Chen, Z., Miao, J., Fang, X., Wu, C.: A real-time detecting algorithm for tracking community structure of dynamic networks. *arXiv preprint arXiv:1407.2683* (2014)
22. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *Proc. of the 4th ICISSP*. pp. 108–116 (2018)
23. Weir, C., Arantes, R., Hannon, H., Kulseng, M.: Operationally Transparent Cyber (OpTC), Data retrieved from IEEE Dataport

Désobfuscation par intelligence artificielle : comment s'en protéger?

Grégoire Menguy¹, Sébastien Bardin¹,
Richard Bonichon² et Cauim de Souza Lima¹

¹ CEA LIST {prénom}.{nom}@cea.fr

² Nomadic Labs richard.bonichon@nomadic-labs.com

Abstract. L’obfuscation de code tente de protéger les secrets contenus au sein des programmes. Récemment, des méthodes boîte noire, basées sur de l’intelligence artificielle, ont été appliquées pour outrepasser les systèmes d’obfuscation standards. Ces méthodes ne semblent pas impactées par les obfuscations usuelles et aucune protection efficace n’a encore été développée. Nous proposons d’approfondir la compréhension de ces méthodes en montrant qu’elles sont bien insensibles aux protections usuelles et en introduisant les premières protections anti-désobfuscation boîte noire. Nous montrons que ces protections sont efficaces contre les débfuscateurs boîte noire (Syntia et Xyntia) à l’état de l’art.

Keywords: Obfuscation · Désobfuscation · Intelligence Artificielle.

1 Introduction

Les programmes peuvent intégrer des informations précieuses comme des algorithmes propriétaires ou des clés cryptographiques. Des attaquants peuvent tenter de les extraire pour outrepasser des protections, porter atteinte à la propriété intellectuelle ou analyser le code. Le scénario *Man-At-The-End (MATE)* considère ainsi que l’utilisateur lui-même est un attaquant essayant d’extraire ces informations secrètes. L’*obfuscation de code* [6] permet de répondre à ce défi en transformant un programme P en un programme équivalent P' plus complexe à comprendre ou modifier (en temps ou en argent). Face à cela, des méthodes dites de *désobfuscation* tentent de récupérer un code proche de l’original à partir de sa version obfusquée. De nombreuses méthodes de désobfuscation ont été proposées. Nombre d’entre elles reposent sur de l’analyse symbolique et ont été prouvées très efficaces contre les stratégies d’obfuscation standards [15, 2, 20]. Cependant, celles-ci sont par nature fortement influencées par la complexité syntaxique du code sous analyse. Cette limitation a mené au développement de nouvelles contre-mesures efficaces [12, 13, 21, 6].

Intelligence artificielle pour la désobfuscation. Les méthodes d’intelligence artificielle (IA) ne sont souvent ni correctes (pouvant retourner de faux résultats) ni complètes (pouvant ne pas trouver de résultats). Cependant, elles permettent de trouver de bons résultats rapidement à des problèmes complexes. Ces

méthodes ont notamment été utilisées en analyse de binaires par Blazytko et al. [3] et Menguy et al. [11]. Ces articles montrent comment des algorithmes de recherche stochastiques [4, 17] peuvent permettre de déobfusquer des expressions protégées. Ils considèrent le (morceau de) code sous analyse comme une boîte noire ce qui leur permet d’outrepasser les protections actuelles. Ces méthodes permettent notamment de contourner les protections mises en place par des outils d’obfuscation tels que VMProtect [18], Themida [14] et Tigress [5].

Problème et Objectif. Comprendre comment se protéger de telles attaques est encore un problème ouvert. Pour pallier cela, nous montrons que les protections à l’état de l’art [21, 7, 19, 12, 16] ne permettent pas de se protéger efficacement et nous proposons les 2 premières protections efficaces contre ces méthodes. Nous évaluons ces protections contre Syntia [3] et Xyntia [11] les deux outils de l’état de l’art. Nous montrons que même si Xyntia est plus efficace (rapide et robuste) que Syntia, ces deux outils sont incapables d’analyser des codes obfusqués via nos méthodes. Ce document se base sur l’article *Search-based Local Blackbox De-obfuscation: Understand, Improve and Mitigate* [11] des même auteurs et accepté pour publication à la conférence ACM CCS 2021.

2 Contexte

2.1 Obfuscation de code

L’obfuscation [6] est une famille de méthodes dont l’objectif est de rendre la rétro-ingénierie d’un programme (comprendre son fonctionnement) difficile. Ces méthodes sont utilisées par les industriels pour protéger leur propriété intellectuelle mais également par les développeurs de logiciels malveillants pour mettre à mal les analyses. Pour ce faire, les méthodes d’obfuscation transforment un programme P en un programme équivalent P' plus complexe. Le but est de maximiser le niveau de complexité du programme sans que cela n’influence trop ses performances. Il est important de comprendre que l’obfuscation ne peut pas réellement empêcher un utilisateur d’analyser le fonctionnement d’un programme – c’est impossible dans le scénario *MATE* [1]. L’objectif est donc de rendre la tâche suffisamment complexe pour dépasser les compétences de l’attaquant. Nous présentons ci-dessous différentes méthodes d’obfuscation. Supposons que nous

```

1 // 2020-08-30 02:00:00
2 const time_t T = 1598745600;
3
4 void foo(void)
5 {
6     time_t t = time(NULL);
7     if (t - T > 0) {
8         malicious();
9     }
10 }
```

Listing 1.1: Code non obfusqué

```

1 const time_t T = 1598745600;
2
3 void foo(void){
4     time_t t = time(NULL);
5     if (t*T + T == 0)
6         complex_dumb1();
7     else if (t - T > 0 && t + T > t)
8         malicious();
9     else complex_dumb2();
10 }
```

Listing 1.2: Prédicats opaques

voulions protéger le code présenté en Listing 1.1. Celui-ci exécute une charge malveillante uniquement s'il est lancé après le 30 août 2020 à 2h. Dans l'état actuel, ce code est simple à comprendre. Comment le rendre plus complexe ?

Les prédicats opaques [7] obfusquent le flot de contrôle du programme en injectant des conditions. Ces conditions sont des tautologies et seule une des deux branches est réellement atteignable. Cette protection est difficile à gérer pour un humain mais est sensible aux méthodes d'exécution symbolique [2]. Le Listing 1.2, présente deux prédicats opaques – ligne 5 et 7. Les fonctions `complex_dumb1` et `complex_dumb2` ne sont en réalité jamais appelées mais l'attaquant pourrait perdre du temps à les analyser.

L'encodage *Mixed-Boolean Arithmetic (MBA)* [21] propose de transformer une expression arithmétique et/ou booléenne en une expression équivalente combinant des opérateurs arithmétiques et booléens. Par exemple, l'expression $x + y$ peut être remplacée par $(x \vee 2y) \times 2 - (x \oplus 2y) - y$ qui est équivalente mais plus complexe. Notez que ces transformations peuvent être appliquées successivement pour augmenter la complexité syntaxique de l'expression. De plus, vérifier l'équivalence d'expressions *MBA* est difficile pour des solveurs SMT [9]. Ainsi, l'encodage *MBA* empêche la simplification automatique de code extrait via des méthodes symboliques. Appliquer cet encodage permet d'obtenir le Listing 1.3. Les conditions lignes 5, 6 et 7 ont été encodées ce qui cache leur sémantique.

```

1  const time_t T = 1598745600;
2
3  void foo(void){
4      time_t t = time(NULL);
5      if (2*(t*t | T) - (t*t ^ T) == 0) complex_dumb1();
6      else if ((t | -2*T) * 2 - (t ^ -2*T) + T > 0 &&
7              (t ^ T) - (~ (2*(t & T))) - 1 > t) {
8          malicious();
9      } else complex_dumb2();
10 }
```

Listing 1.3: Code obfusqué

La virtualisation [19] propose de traduire le code écrit par le développeur en *bytecodes* et d'intégrer une machine virtuelle (VM) pour les exécuter. Chaque *bytecode* est associé à un *handler*, stocké dans la table de *handlers* et chargé d'exécuter l'opération voulue (+, −, ×, ∧, ∨, ⊕). Le code résultant applique 3 étapes successives: 1. récupération du *bytecode* à exécuter; 2. décodage du *bytecode* (i.e., récupération dans la table des handlers du *handler* associé); 3. exécution du *handler* sur les entrées spécifiées par le *bytecode*. Par exemple, considérons le *bytecode* `ADD X, 1`, la VM va récupérer le handler *h* associé à `ADD` dans la table des handlers puis exécutera $h(X, 1)$. Évidemment, dans un programme réel, on ne retrouvera pas le symbole `ADD` dans le *bytecode*. Cela rendrait le reverse engineering trop simple. Ainsi, pour comprendre les *bytecodes* il faut analyser chaque handler pour savoir ce qu'ils calculent (dans notre exemple, il faut analyser *h* pour comprendre qu'il réalise une addition). Cependant, les handlers peuvent eux aussi être obfusqués. Néanmoins, même dans ce cas plus complexe, cette protection a été prouvée vulnérable aux approches en boîte noire [3, 11].

2.2 Désobfuscation de code

La désobfuscation a pour objectif d’extraire une version simplifiée d’un programme obfusqué. De nombreuses méthodes reposant sur de l’*exécution symbolique* sont très efficaces contre des protections standards comme les prédicats opaques [15, 2, 20]. Cependant, des méthodes anti-symboliques existent limitant leurs usages [12, 13, 16, 19].

Désobfuscation boîte noire. Blazytko et al. [3] puis Menguy et al. [11] ont proposé respectivement Syntia et Xyntia,³ deux outils de désobfuscation boîte noire reposant sur de l’intelligence artificielle (MCTS [4] et ILS [10]) pour simplifier des blocs de code hautement obfusqués. Pour fonctionner, ces méthodes doivent connaître 1. la *reverse window* (i.e. le bloc de code à analyser); 2. les entrées et sorties de cette *reverse window*. Comme illustration, considérons le Listing 1.4. Pour comprendre la condition en ligne 4, l’analyste se concentre sur le code entre les lignes 1 et 3: c’est notre *reverse window*. L’analyste doit alors déterminer ses entrées et sorties pertinentes. Ici la condition est réalisée sur **t3**: c’est notre sortie. Les entrées correspondent alors à toute variable (registres ou zones mémoire en assembleur) pouvant influencer la valeur de **t3**. Ici, les entrées sont **x** et **y**. Grâce à ces informations, Syntia et Xyntia exécutent le code sur des entrées échantillonnées aléatoirement pour en observer les sorties. Ces échantillons pourraient être $(x \mapsto 1, y \mapsto 2)$, $(x \mapsto 0, y \mapsto 1)$ et $(x \mapsto 3, y \mapsto 4)$ donnant les sorties 3, 1 et 7 respectivement. Syntia et Xyntia synthétisent alors une expression mimant les comportements observés. Ici, ils synthétisent $t3 \leftarrow x + y$. L’analyste peut alors conclure que la condition est $x + y = 5$. Ceci est bien plus simple et exploitable que le résultat d’une méthode symbolique, typiquement $((x \vee 2y) \times 2 - (x \oplus 2y) - y) = 5$.

```

1 int t1 = 2 * y;
2 int t2 = x | t1;
3 int t3 = t2 * 2 - (x ^ t1) - y;
4 if (t3 == 5) ...

```

Listing 1.4: Obfuscated condition

3 Protections Usuelles

La désobfuscation boîte noire utilise des couples entrées-sorties pour synthétiser la sémantique d’un bloc de code. Elle ne devrait donc pas être influencée par les protections usuelles qui modifient uniquement la syntaxe du code. Nous nous proposons de le vérifier en évaluant Syntia et Xyntia contre différentes méthodes d’obfuscation à l’état de l’art: l’encodage MBA [21], les prédicats opaques [7], l’obfuscation via explosion de chemin [19, 12], les *covert channels* [16]. Pour ce faire, nous avons obfusqué avec Tigress [5] le *dataset* introduit par Menguy et al. [11] pour évaluer Syntia et Xyntia. Nous évaluons l’impact des protections sur le taux de réussite (pourcentage d’expressions synthétisées avec succès).

³ intégré dans le framework d’analyse de binaire Binsec [8] et accessible à l’adresse <https://github.com/binsec/xyntia>

	\emptyset	MBA	Opaque	Path oriented	Covert channels
Syntia	34.5%	31.4%	33.9%	33.4%	34.1%
Xyntia	95.5%	95.4%	94.68%	95.4%	95.1%

Table 1: Taux de réussite de Syntia et Xyntia en fonction des protections utilisées (\emptyset = pas de protection)

Résultats. Comme prévu (Table 1), ni Syntia ni Xyntia ne sont perturbés par l’encodage MBA, les prédicats opaques ou l’obfuscation via explosion de chemin. Le cas des *covert channels* est plus intéressant. En effet, cette protection, modifie la sémantique du code en introduisant une probabilité d’erreur. Ainsi, avec une faible probabilité, le code protégé ne se comportera pas comme le programme original. Cela pourrait avoir une incidence sur Syntia et Xyntia en faussant la phase d’échantillonnage. Cependant, nos expériences montrent que la probabilité d’erreur est si faible que cela est sans conséquence sur Syntia ou Xyntia.

4 Anti-deobfuscation boîte noire

Les protections usuelles étant inefficaces, nous proposons ici 2 nouvelles protections efficaces contre la désobfuscation boîte noire. Celles-ci augmentent non pas la *complexité syntaxique* mais la *complexité sémantique* du code sous analyse. Nous nous plaçons dans le contexte de la virtualisation qui a déjà été prouvé vulnérable aux approches en boîte noire [3, 11].

Problème et Objectifs. Pourquoi la virtualisation est-elle vulnérable aux méthodes en boîte noire? Cela est due au fait que les handlers sont bien délimités (on trouve rapidement le début et la fin d’un handler) et réalisent des opérations sémantiquement simples – par exemple, $+$, $-$, \times , \wedge . Ainsi, il est aisé pour l’analyste de définir la *reverse window* – dans notre cas, le handler complet – et le synthétiseur ne rencontrera pas de problème pour inférer la sémantique du code. Ainsi, pour protéger les handlers, il est nécessaire d’augmenter leur complexité sémantique pour empêcher la synthèse. Nous proposons donc 2 nouvelles protections: les *handlers sémantiquement complexes* et les *handlers fusionnés*.

4.1 Handlers sémantiquement complexes

Cette protection consiste à générer des handlers sémantiquement complexes tout en conservant un ensemble de handlers Turing-complet.

Définition. Soit S un ensemble d’expression et h, e_1, \dots, e_{n-1} n expressions dans S . Supposons que (S, \star) forme un groupe. On note $-e_i$ l’inverse de e_i dans (S, \star) .

Alors h peut être encodé comme suit, $h = \bigstar_{i=0}^{n-1} h_i$ où pour tout $0 \leq i < n$,

$$h_i = \begin{cases} h - e_1 & \text{if } i = 0 \\ e_i - e_{i+1} & \text{if } 1 \leq i < n - 1 \\ e_{n-1} & \text{if } i = n - 1 \end{cases}$$

Ainsi, chaque h_i est un nouvel handler pouvant être combiné avec d'autres pour calculer des opérations simples. Par exemple, $h = x + y = h_0 + h_1 + h_2$ où $h_0 = (x + y) - ((a - x^2) - (xy))$, $h_1 = (a - x^2) - xy + (-(y - (a \wedge x)) \times (y \otimes x))$ et $h_2 = (y - (a \wedge x)) \times (y \otimes x)$. Notez que le choix des e_i est arbitraire. Ils peuvent donc être aussi complexe que voulu.

Évaluation expérimentale. Pour évaluer notre approche, nous avons généré 3 datasets de handlers, BP1, BP2 et BP3 classés par ordre croissant de complexité. Chaque dataset contient 15 handlers sémantiquement complexes permettant d'encoder 5 opérations simples: $+$, $-$, \times , \wedge , \vee . Au sein d'un dataset, tous les handlers considèrent le même nombre d'entrées.

Résultats. Nos expériences montrent que Xyntia (avec un timeout de 1h par handler) est capable d'inférer 13/15 handlers de BP1. Cependant, les performances se dégradent très rapidement (BP2: 3/15, BP3: 1/15). Syntia en revanche ne gère efficacement aucun dataset. Même avec un timeout de 12h par handler, Syntia ne synthétise que 1/15 handler de BP1 et aucun de BP2 et BP3. Ainsi, avec une complexité moyenne (BP2) nous sommes capables de protéger efficacement les handlers.

Discussion. Notre protection empêche les méthodes boîte noire de synthétiser chaque handler séparément. Néanmoins, elle peut être contournée si l'analyste trouve le bon ensemble de handler à synthétiser. Pour éviter cela, les handlers peuvent être dupliqués – comme dans VMProtect [18] – pour complexifier une recherche par motif.

4.2 Handlers fusionnés

Cette protection consiste à fusionner deux handlers en un unique via une condition *if-then-else* (ITE). Par exemple, les deux handlers $h_1(x, y) = x + y$ et $h_2(x, y) = x \times y$ peuvent être fusionnés en $h(x, y, c) = \text{if}(c = 10) \text{ then } h_1(x, y) \text{ else } h_2(x, y)$. Il est nécessaire de noter qu'en pratique nous n'utiliserons pas de conditionnelles usuelles (introduisant des branchements conditionnels au niveau assembleur). En effet, l'analyste pourrait simplement retrouver h_1 et h_2 en se concentrant sur les deux blocs séparément. Ainsi, nous utilisons un encodage sans branchement comme présenté en Fig. 1. Nous supposons donc que l'analyste voit un unique bloc et ne peut pas analyser les branches séparément.

```
// if (c == cst) then h1(a,b,c) else h2(a,b,c);
int32_t res = c - cst;
res = (-((res ^ (res >> 31)) - (res >> 31)) >> 31) & 1;
return h1(a, b, c)*(1 - res) + res*h2(a, b, c);
```

Fig. 1: Example of a branch-less condition

Évaluation. Pour évaluer notre approche nous considérons 5 datasets contenant 20 handlers chacun. Les handlers dans le dataset 1 contiennent un ITE combinant 2 handlers atomiques. Les handlers du second dataset contiennent 2 ITE imbriqués (3 handlers atomiques) et ainsi de suite. Les conditions vérifie l'égalité d'une variable par rapport à une constante. Par exemple, le dataset 2 contient le handler $ITE(c = 0, x + y, ITE(c = 1, x - y, x \times y))$. Pour évaluer Xyntia et Syntia nous avons considéré différents scénario. Notamment le scénario *Utopique* ou Xyntia et Syntia possède la bonne grammaire d'expressions, les ITEs adéquats et les constantes utiles. De plus, l'échantillonnage des entrées est réalisé de tel sorte que toutes les branches sont traversées le même nombre de fois. Ce scénario avantage beaucoup l'analyse et permet de montrer que même dans ce cas, la désobfuscation boîte noire ne peut pas passer notre protection.

Résultats. Nos expériences montrent que, même dans le scénario *Utopique*, Xyntia n'est pas capable de gérer des handlers contenant ≥ 3 ITE imbriqués (i.e. 4 branches). Syntia quant à lui, n'est pas capable de gérer des handlers contenant ≥ 2 ITE imbriqués.

Discussion. La désobfuscation symbolique n'est pas touchée par cette protection. Ainsi, pour se protéger efficacement, il est nécessaire de combiner notre approche avec des protections anti-exécution symbolique.

5 Conclusion

Xyntia et Syntia sont des déobfuscateurs boîte noire reposant sur des méthodes d'intelligence artificielle. Comme nous l'avons montré ces méthodes ne sont pas sensibles aux protections usuelles. Nous avons donc proposé deux nouvelles protections capable de se protéger efficacement contre les méthodes boîte noire. Notre évaluation montre que ni Syntia ni Xyntia – les deux déobfuscateurs boîte noire à l'état de l'art – ne sont capables d'écarter nos protections.

References

1. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im) possibility of obfuscating programs. *Journal of the ACM (JACM)* **59**(2), 1–48 (2012)
2. Bardin, S., David, R., Marion, J.: Backward-bounded DSE: targeting infeasibility questions on obfuscated codes. In: 2017 IEEE Symposium on Security and Privacy, SP 2017. IEEE Computer Society
3. Blazytko, T., Contag, M., Aschermann, C., Holz, T.: Syntia: Synthesizing the semantics of obfuscated code. In: *USENIX Security*. pp. 643–659 (2017)
4. Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* **4**(1), 1–43 (2012)
5. Collberg, C., Martin, S., Myers, J., Zimmerman, B.: The Tigress C Diversifier/Obfuscator, <http://tigress.cs.arizona.edu/>

6. Collberg, C., Nagra, J.: *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Addison-Wesley Professional, 1st edn. (2009)
7. Collberg, C., Thomborson, C., Low, D.: Manufacturing cheap, resilient, and stealthy opaque constructs. In: *Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. pp. 184–196 (1998)
8. David, R., Bardin, S., Ta, T.D., Mounier, L., Feist, J., Potet, M.L., Marion, J.Y.: Binsec/se: A dynamic symbolic execution toolkit for binary-level analysis. In: *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. vol. 1, pp. 653–656. IEEE (2016)
9. Eyrolles, N., Goubin, L., Videau, M.: Defeating mba-based obfuscation. In: *Proceedings of the 2016 ACM Workshop on Software PROtection, SPRO@CCS 2016* (2016)
10. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search: Framework and applications. In: *Handbook of metaheuristics*, pp. 129–168. Springer (2019)
11. Menguy, G., Bardin, S., Bonichon, R., Lima, C.d.S.: Search-based local blackbox deobfuscation: Understand, improve and mitigate. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021)
12. Ollivier, M., Bardin, S., Bonichon, R., Marion, J.Y.: How to kill symbolic deobfuscation for free (or: unleashing the potential of path-oriented protections). In: *35th Annual Computer Security Applications Conference* (2019)
13. Ollivier, M., Bardin, S., Bonichon, R., Marion, J.Y.: Obfuscation: where are we in anti-dse protections?(a first attempt). In: *Proceedings of the 9th Workshop on Software Security, Protection, and Reverse Engineering*. pp. 1–8 (2019)
14. Oreans Technologies: Themida – Advanced Windows Software Protection System. <http://oreans.com/themida.php> (2020)
15. Schrittwieser, S., Katzenbeisser, S., Kinder, J., Merzdovnik, G., Weippl, E.: Protecting software through obfuscation: Can it keep pace with progress in code analysis? *ACM Comput. Surv.* **49**(1) (2016)
16. Stephens, J., Yadegari, B., Collberg, C.S., Debray, S., Scheidegger, C.: Probabilistic obfuscation through covert channels. In: *2018 IEEE European Symposium on Security and Privacy, EuroS&P, 2018*
17. Talbi, E.G.: *Metaheuristics: From Design to Implementation*. Wiley Publishing (2009)
18. VM Protect Software: VMProtect Software Protection. <http://vmpsoft.com> (2020)
19. Yadegari, B., Debray, S.: Symbolic execution of obfuscated code. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15*, ACM
20. Yadegari, B., Johannesmeyer, B., Whitely, B., Debray, S.: A generic approach to automatic deobfuscation of executable code. In: *Symposium on Security and Privacy, SP* (2015)
21. Zhou, Y., Main, A., Gu, Y.X., Johnson, H.: Information hiding in software with mixed boolean-arithmetic transforms. In: *Proceedings of the 8th International Conference on Information Security Applications*. pp. 61–75. WISA'07, Springer-Verlag, Berlin, Heidelberg (2007)

Multi-fidelity constrained Bayesian optimization, application to drone design

Rémy Charayron^{1,2}, Thierry Lefebvre¹, Nathalie Bartoli¹, and Joseph Morlier^{2,3}

¹ ONERA/DTIS, Université de Toulouse, Toulouse, France

² ISAE-SUPAERO, Toulouse, France

³ ICA, Université de Toulouse, INSA, CNRS, MINES ALBI, UPS, Toulouse, France

Abstract. In aeronautics, the first design stages usually involve to solve a constrained multi-disciplinary optimization problem. The Bayesian optimization strategy is a way to solve such a complex system. This approach requires to evaluate the objective function and the constraints quite a few times. Evaluations are generally performed using numerical models that can be computationally expensive. To alleviate the overall optimization cost variable information sources can be used to make the evaluations. Typically we are dealing with cheap low fidelity models to explore the design space and expensive high fidelity models for exploitation. In the following work, a mono-fidelity Bayesian optimization method and its multi-fidelity counterpart are compared on two analytical test cases and on an aerostuctural drone design constrained optimization problem. The multi-fidelity strategy allows to divide the computational cost by 1.3 compared to the mono-fidelity one on these test cases.

Keywords: Drone design · Multi-disciplinary optimization · Constrained Bayesian optimization · Variable fidelity · Surrogate models · Kriging · Gaussian process.

1 Introduction

In the first design steps, drone design optimization relies on multidisciplinary numerical models. These models capture the interactions between the different disciplines (aerodynamic, structure, operations, ...) which play a role in the drone overall performance. It follows that a single evaluation of the model is computationally expensive. Moreover, the complexity of the coupled system does not encourage us to take advantage of the analytical gradients. The high computational cost of a single evaluation implies that finite differences or complex step methods traditionally used in order to approximate the gradient can not be considered, hence classical gradient based optimization methods can not be used and the model is considered as a black-box function for which no information (regularity properties, derivative, ...) are available. Similarly the use of evolutionary optimization algorithms is not allowed due to the large number of function evaluations required. Then the focus is made on gradient-free surrogate-based optimization methods [12]. This involves to replace the initial model to optimize with a cheaper one, called metamodel or surrogate model. To construct this surrogate, gaussian processes (GP) [25] [19] interpolation framework also called kriging [21] is very powerful. Indeed, it allows not only to provide a prediction (which is the mean of the GP) but also the uncertainty of this prediction (the variance of the GP). In fact, the approach takes advantage of the gaussian vectors conditional distribution in order to determine the posterior distribution of the GP knowing some realizations of the initial model called design of experiments (DoE). This gives us a first surrogate model to approximate the initial model. Then it is improved via an iterative process that adds observations to the DoE according to a certain rule that tries to define the most interesting point to evaluate at each iteration doing a trade-off between exploitation and exploration. This rule is called the acquisition function and the whole process defined Bayesian optimization (BO) methods [13] whose first implementation was EGO in [16]. When the cost of evaluating the initial model is so important that even the previous BO method becomes intractable, multi-fidelity BO methods can be useful. These kind of methods use various levels of code, the highest fidelity (*HF*) code being the initial model and the lowest ones being some cheapest to evaluate approximations of the *HF* model. The benefit of multi-fidelity BO methods is that, depending on the case, the evaluation at the point to add to the DoE can be done using different codes: with a very precise but very expensive one or with less precise but less expensive codes. In Section 2 the kriging and Bayesian optimization methodology are

introduced. Section 3 focuses on the extension of this methodology to multi-fidelity. Section 4 and Section 5 present respectively some analytical test cases and a drone design case in order to illustrate and validate the proposed approach.

2 State of the Art

Let s the function defined in Eq. (1) that can only be evaluated in order to be optimized

$$\begin{aligned} s : \Omega \subset \mathbb{R}^d &\rightarrow \mathbb{R} \\ x &\mapsto y = s(x) \end{aligned} \quad (1)$$

2.1 Gaussian Processes interpolation / kriging method

The function s is considered to be the realization of a gaussian process $Z \sim GP(\mu, k)$ with prior mean $\mu : \Omega \rightarrow \mathbb{R}$ and covariance kernel $k : \Omega^2 \rightarrow \mathbb{R}$. The covariance kernel has $d + 1$ hyperparameters $\theta = \{\sigma^2 = \theta_0, (\theta_i)_{i=1, \dots, d}\}$. The overall prior variance σ^2 is a scaling factor. The θ parameters are the correlation lengths in each direction. Let suppose that l observations of the function s are gathered in a DoE $D = \{x_k, y_k\}_{k=1, \dots, l}$ where $x_k \in \Omega$ and $y_k = s(x_k)$. The GP conditioned by D defines for each point $x \in \Omega$ a random variable y_x^D which follows a gaussian distribution

$$y_x^D \sim \mathbb{P}(y|(D, x)) = GP((\mu, k)|(D, x)) = \mathcal{N}(\hat{\mu}(x), (\hat{\sigma}(x))^2)$$

If no information on the GP mean μ is available, it is assumed to be unknown, then μ is supposed to be the zero constant function: we talk about ordinary kriging. Else, if there is a known trend in the data, it can be modeled using a deterministic basis of functions. The prior mean of the GP at a point x can be written as:

$$\mu(x) = \sum_{k=1, \dots, p} \beta_k f_k(x) \quad (2)$$

with f_k the k -th basis function and β_k the coefficient associated to the k -th basis function. In this case we talk about universal kriging. Lets denote $\boldsymbol{\mu} = (\mu(x_0) \dots \mu(x_l))^T$, $\mathbf{k}(x) = (k(x_0, x) \dots k(x_l, x))^T$, $\mathbf{Y} = (y_0 \dots y_l)^T$ and $\mathbf{K} = \left(k(x_i, x_j) \right)_{i,j=1, \dots, l}^T = \sigma R$ the covariance matrix on all the sampling points (\mathbf{K} depends on θ) where R is the correlation matrix. Then using the gaussian vector conditional rule, it follows the subsequent expressions:
$$\begin{cases} \hat{\mu}(x) = \mu(x) + \mathbf{k}^T \mathbf{K}^{-1} (\mathbf{Y} - \boldsymbol{\mu}) \\ \hat{\sigma}(x) = (k(x, x) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k})^{\frac{1}{2}} \end{cases}$$

The posterior mean $\hat{\mu}$ represents the surrogate model that approximates our function s . An indication on this surrogate model accuracy is given by the posterior standard deviation $\hat{\sigma}$. Note that the kriging kind of surrogate model has been selected specifically because it allows to have the variance expression on all the design space. This information is crucial using a Bayesian optimization strategy described in Section 2.2. To know the variance information using other kinds of surrogate models (radial basis function, neural networks, polynomial approximation, ...) would have required to perform a bootstrap method [11]. Therefore several metamodels should have been constructed in parallel to approximate the variance. Associated computation effort would have then led to an almost intractable method. GPs are parameterized by the kernel hyperparameters that need to be estimated. The maximum likelihood estimation method with a likelihood concentration process is used in this goal [23]. Dealing with high dimensional problems when using a kriging method raises additional difficulties. The number of hyperparameters increases with the dimension and their estimation is harder to optimize. One way to tackle these difficulties is the use of Partial Least Squares (PLS) [6] [22]. The PLS method finds a linear relationship between input variables and the output variable by projecting input variables onto a new space of lower dimension. The latent variables are linear combinations of the initial ones. KPLS defines a new covariance kernel which uses a lower number of hyperparameters [6] [7] and will be used in the Bayesian strategy described in the following..

2.2 Bayesian optimization

Bayesian optimization [13] is a global optimization strategy usually applied to optimize expensive to evaluate black-box functions. It consists in building a surrogate model of the objective function and then iteratively enriching this surrogate model with objective function evaluations to explore the design space and ensuring that the surrogate is precise enough in the optimal area.

Unconstrained BO Let the following unconstrained optimization problem:

$$x^* = \arg \min_{x \in \Omega} s(x) \quad \text{with } \Omega \subset \mathbb{R}^d \quad (3)$$

where $s : \Omega \rightarrow \mathbb{R}$ is the objective function introduced in Eq. (1). The efficient global optimization [16] called EGO constructs a GP of the objective function s using an initial DoE. Then the optimal solution is found by enriching iteratively the DoE and the GP. This enrichment is based on a trade-off, the exploration of the design space Ω and the exploitation of the GP model to find the minimum. The strategy involves the resolution of an optimization sub-problem to determine the next point to evaluate. This sub-problem is defined via an acquisition function $\alpha : \mathbb{R}^d \rightarrow \mathbb{R}$ to maximize

$$x_{next} = \arg \max_{x \in \Omega} \alpha(x) \quad (4)$$

There exists an extensive literature on acquisition functions [24] [13]. Some well known criteria are recalled in the following.

- Expected Improvement (*EI*): The *EI* computes the expected improvement of the current minimum value in the DoE by adding an evaluation.

$$\alpha_{EI}(x) = EI(x) = \mathbb{E}(I(x)) = \mathbb{E}(\max(0, f_{\min} - y_x^D)) \quad (5)$$

where $f_{\min} = \min_{x \in X_D} (s(x))$ with X_D is the input set of the DoE D . In the case where y_x^D follows a gaussian law, $y_x^D \sim \mathcal{N}(\hat{\mu}(x), (\hat{\sigma}(x))^2)$, the *EI*(x) criterion is analytical:

$$EI(x) = \alpha_{EI}(x) = \begin{cases} 0 & \text{if } \hat{\sigma}(x) = 0 \\ (f_{\min} - \hat{\mu}(x))\Phi\left(\frac{f_{\min} - \hat{\mu}(x)}{\hat{\sigma}(x)}\right) + \hat{\sigma}(x)\phi\left(\frac{f_{\min} - \hat{\mu}(x)}{\hat{\sigma}(x)}\right) & \text{else} \end{cases} \quad (6)$$

where Φ and ϕ represent respectively the $\mathcal{N}(0, 1)$ cumulative distribution function and the probability density function. The first term of the expression is the exploitation term, it increases when $\hat{\mu}(x)$ decreases while the second term is the exploration term, it increases when the GP is not precise, ie when $\hat{\sigma}(x)$ is large.

- Watson and Barnes 2 (*WB2*): *WB2* criterion [28] tries to regularize the *EI* by adding the mean $\hat{\mu}(x)$:

$$\alpha_{WB2}(x) = \alpha_{EI}(x) + \hat{\mu}(x) \quad (7)$$

Appendix C shows the six first iterations of the BO process on a one dimensional test case using the EGO algorithm.

Constrained BO (CBO) Let the following constrained optimization problem:

$$x^* = \arg \min_{x \in \Omega} s(x) \quad \text{such that } g(x) \geq 0 \quad \text{and } h(x) = 0 \quad (8)$$

where the constraints are defined by

- $g : \mathbb{R}^d \rightarrow \mathbb{R}^m$ (m inequality constraints)
- $h : \mathbb{R}^d \rightarrow \mathbb{R}^p$ (p equality constraints)

The CBO algorithm is quite similar as the one of the unconstrained BO approach except that the optimization sub-problem solved to enrich the DoE takes into account the constraints. The associated sub-problem can take two forms: it can be unconstrained and tries to optimize an adapted function which gathers the constraints and the classical criterion [14]; or it can be constrained and optimizes one of the previous acquisition functions with some feasibility criteria associated to the constraints g and h [4]. Here the focus is made on constrained optimization sub-problem methods. The optimization sub-problem is of the form

$$x_{next} = \arg \max_{x \in \Omega} \alpha(x) \quad \text{with } x \in \Omega_h \cap \Omega_g \quad (9)$$

where Ω_h and Ω_g are respectively the feasible domains defined by the two feasibility criteria: $\alpha_h : \mathbb{R}^d \rightarrow \mathbb{R}^p$ and $\alpha_g : \mathbb{R}^d \rightarrow \mathbb{R}^m$. To construct the feasibility criteria, the approaches named Super Efficient Global Optimization (SEGO) [27] and the Super Efficient Global Optimization coupled with Mixture Of Experts (SEGOMOE) [3] [2] [4] use the posterior means of the GPs that modelize the constraints as feasibility criterion: $\alpha_h = \hat{\mu}_h$ and $\alpha_g = \hat{\mu}_g$. The feasible domains are $\Omega_h = \{x, \alpha_h(x) = 0\}$ and $\Omega_g = \{x, \alpha_g(x) \geq 0\}$.

3 MFSEGO methodology

Using various information sources can be useful to alleviate the computation cost to build an accurate surrogate model or to perform an optimization. The SEGO type approaches are now extended to multi-fidelity and denoted in the following by MFSEGO.

3.1 Multi-fidelity kriging

Making assumptions in order to link the different fidelity levels is a way to simplify multi-fidelity problems. A discrepancy function δ that captures the difference between the high fidelity (HF) and low fidelity (LF) levels and a scaling factor ρ are considered in [17]

$$f_{HF}(x) = \rho f_{LF}(x) + \delta(x) \quad \text{such that} \quad f_{LF} \perp \delta \quad (10)$$

Le Gratiet [20] proposed to add the LF function to the basis function set $(h_i)_{i=1\dots p}$ used in the universal kriging regression term (see Eq. (2)) to get:

$$\mu(x) = \sum_{i=1, \dots, p} \left(\beta_i h_i(x) \right) + \beta_\rho f_{LF}(x) \quad (11)$$

β_ρ is an estimation of ρ done at the hyperparameters estimation step (see Section 2.1). Using a nested DoE structure: $D_{HF} \subseteq D_{LF}$, the independence between the high and low fidelity of the surrogate model is assumed. Then the HF surrogate model mean and variance can be expressed as:

$$\begin{cases} \hat{\mu}_{HF} = \rho \hat{\mu}_{LF} + \hat{\mu}_\delta \\ \hat{\sigma}_{HF}^2 = \rho^2 \hat{\sigma}_{LF}^2 + \hat{\sigma}_\delta^2 \end{cases} \quad (12)$$

Le Gratiet's approach can then be extended to $L+1$ fidelity levels. Let us denote the fidelity levels f_0, \dots, f_L sorted from the lowest to the highest (we still consider a nested DoE structure: $D_L \subseteq D_{L-1} \subseteq \dots \subseteq D_0$). The following recursive formulation can be written $\forall k = 1, \dots, L$:

$$\begin{cases} \hat{\mu}_k = \rho_{k-1} \hat{\mu}_{k-1} + \hat{\mu}_{\delta_k} \\ \hat{\sigma}_k^2 = \rho_{k-1}^2 \hat{\sigma}_{k-1}^2 + \hat{\sigma}_{\delta_k}^2 \end{cases} \quad (13)$$

In this case, ρ is considered as a constant but it can depends on x . Then we have $\rho : x \mapsto \rho(x)$. This have been implemented in the toolbox SMT [8].

To learn the multi-fidelity model, the lowest fidelity level is learnt first, then the relationships (scaling factor ρ and discrepancy function δ) between every successive fidelity level are consecutively learnt. Since the variances can be expressed in closed form, the contribution of each fidelity level to the total variance of the multi-fidelity model can be deduced too. Denoting $\sigma_{cont}^2(k, x)$ the variance contribution of the k^{th} fidelity level at the point x , with the notation $\sigma_{\delta_0}^2 = \sigma_0^2$ and assuming that $\prod_{j=k}^{L-1} \rho_j^2 = 1$, we have:

$$\sigma_{cont}^2(k, x) = \sigma_{\delta_k}^2(x) \prod_{j=k}^{L-1} \rho_j^2 \quad (14)$$

3.2 Multi-fidelity Bayesian optimization

With a multi-fidelity Bayesian optimization process, when a point is added to the DoE, not only the most promising point has to be decided, but also the fidelity level to which evaluate it. Splitting the problem of finding the point and the fidelity level in two successive steps has been proposed in [20]. First the point is found using a classical acquisition function as in the mono-fidelity Bayesian optimization (see Section 2.2). Then the variance contribution knowledge at each fidelity level gives some information to smartly decide the fidelity level to choose. The principal advantage of the multi-fidelity Kriging formulation presented in Section 3.1 lies in the fact that the variance contribution of each fidelity level can be known analytically. On the other side the main drawback is that it requires a nested DoE structure.

Let c_0, \dots, c_L be respectively the querying costs of all the fidelity levels f_0, \dots, f_L . Let us denote $\sigma_{red}^2(k, x^*)$ the variance reduction of the high fidelity model when the point x^* is evaluated with all the fidelity levels $\leq k$

$$\sigma_{red}^2(k, x^*) = \sum_{i=0}^k \sigma_{\delta_i}^2(x^*) \prod_{j=i}^{L-1} \rho_j^2 \quad (15)$$

A criterion to choose the level of enrichment can be written as

$$t = \arg \max_{k \in \{0, \dots, L\}} \frac{\sigma_{red}^2(k, x^*)}{(\sum_{i=0}^k c_i)^2} \quad (16)$$

This two step approach MFSEGO combining Eq. (9) and Eq. (16) is described in a pseudo-code in Appendix E and in Figure 5 of the Appendix F. It is now applied on different test cases and compared to SEGO.

4 Analytical cases

To start, the SEGO and MFSEGO methods have been confronted on two analytical test cases. The Branin and Sasena test cases are two different problems with a 2D objective function ($\Omega_{Branin} = [0; 1]^2$) and $\Omega_{Sasena} = [0; 5]^2$ respectively) with a single constraint. The cost ratio between high and low fidelity is arbitrarily fixed to $\frac{cost_{HF}}{cost_{LF}} = 5$. The tolerance on the constraints and the reference solutions for these two analytical test cases are detailed in Table 1. The HF and LF expressions of the objective function and constraint for the Branin and Sasena cases are given in Appendices A and B.

Table 1: Tolerance on the constraints and the reference solutions for the Branin and Sasena cases

	optimal objective value	ref _{sol}	ϵ	tol constraint
Branin	24.863	(0.498, 0.401)	0.5%	$1e^{-4}$
Sasena	-1.172	(2.745, 2.352)	0.5%	$1e^{-4}$

For all the test cases, 10 SEGO runs and 10 MFSEGO runs were made. Mono-fidelity and multi-fidelity runs share the same initial HF DoE. For the multi-fidelity runs, a LF DoE twice the size of the HF DoE is added. For all the optimization runs the squared exponential kernel with a constant trend is chosen to build

the kriging model. The size of the initial DoE and the budget are summed up in Table 3 for each test case. Given the reference solution, a convergence criterion for the mean (over the 10 runs) of the objective value at the best valid point \bar{y}_{best}^{valid} is defined by:

$$\frac{\bar{y}_{best}^{valid} - f_{HF}(\text{ref}_{sol})}{f_{HF}(\text{ref}_{sol})} \leq \epsilon \quad (17)$$

For both mono and multi-fidelity strategies, the number of HF iterations and the total cost required to reach the convergence criterion (see Eq. (17)) are compared for each test case. Results are summed up in Tables 4 and 5 and convergence illustrations are given in Figure 3. Note that for the Branin case it is possible to go slightly below the reference solution because a tolerance on the constraint is considered. For Sasena test case, in the mono-fidelity case, the budget allocated to the simulation is not enough to reach the convergence criterion from Eq. (17). This issue can be explained by the fact that a run did not converge to the global minimum. On Figure 3(b) the median of the best valid value at each HF iteration is considered. Unlike the mean, the median converges even when the optimization is performed with mono-fidelity. Even if the convergence rate of the median is quite the same for mono and multi-fidelity, the multi-fidelity approach seems more robust. In the end, to reach the convergence criterion of the best valid objective mean value for the Branin and the Sasena test case, MFSEGO methodology allows to divide the SEGO total cost by 1.25 and 1.40 respectively.

5 Drone design test case

With the goal of designing a fixed-wing drone, a drone design test case is introduced. It relies on the K75 of Elecnor Deimos, a drone from specific class of about 80 kg MTOW and 35 kg payload mass available at ONERA and illustrated on Figure 1. The K75 delivered to ONERA in April 2018 is the first in the series, it is commercialized by Elecnor Deimos under the name D80-Titan [1, 5].



Fig. 1: Illustration of a K75 drone.

The focus is made on the aerostructure part which involves two disciplines, aerodynamic and structure. Two aerostructural models were developed for the K75 using OpenAeroStruct (OAS) [15] [10] [9], each one associated to a different discretization of the wing and tail meshes (see Table 2 and Figure 2). In our models, two flight points were considered: cruise flight (load factor = 1) and maneuvering (load factor = 9). The HF and the LF models have been evaluated on 200 points chosen randomly in the design space in order to make an estimation of the cost ratio: $\frac{cost_{HF}}{cost_{LF}} = 4.27$.

Table 2: HF and LF mesh dimensions

	HF wing mesh	HF tail mesh	LF wing mesh	LF tail mesh
Chordwise dim	5	5	3	2
Spanwise dim	25	13	9	5

The K75 problem to solve is the following:

$$\begin{aligned} \min_{x \in \Omega} \quad & \text{wing}_{\text{mass}}(x) \quad x \in \mathbb{R}^{15} \\ \text{such that} \quad & \begin{cases} \text{CL}_{\text{cruise}} = 0.5 & \text{CL}_{\text{maneuver}} = 0.5 \\ \text{wing}_{\text{failure, cruise}} \leq 0 & \text{wing}_{\text{failure, maneuver}} \leq 0 \\ \text{tail}_{\text{failure, cruise}} \leq 0 & \text{tail}_{\text{failure, maneuver}} \leq 0 \\ \text{wingbox}_{\text{volume}} - \text{fuel}_{\text{volume}} > 0 & \frac{\text{fuel}_{\text{mass}} - \text{fuel}_{\text{burn}}}{\text{fuel}_{\text{mass}}} = 0 \end{cases} \end{aligned}$$

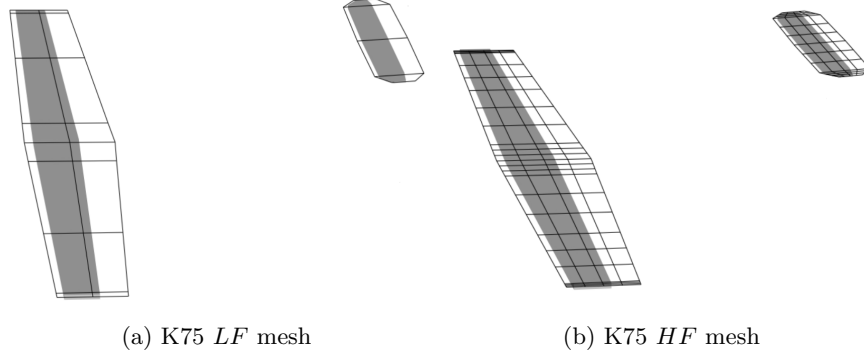


Fig. 2: *LF* and *HF* meshes of the K75

The objective function is the mass of the wing. The 15 design variables and their associated bounds are summarized in Table 6 in Appendix D. Eight constraints are considered: two constraints to ensure a certain lift coefficient value in cruising flight as in maneuvering, four failure constraints to ensure that neither the wing nor the tail will break, whether in cruising flight or during maneuvering. These constraints compare the Von mises stress to the yield stress divided by a fixed coefficient (chosen here equal to 2.5) that acts as a safety margin. To simplify the optimization problem, the individual nodal failure constraints are aggregated using a Kreisselmeier-Steinhauser (KS) function [29]. Next, a constraint that ensures that the wingbox has enough internal volume for the fuel and a constraint that ensures that the fuel burn is equal to the fuel mass are added. The tolerances on these constraints are fixed to 10^{-3} except for the failure constraints for which the tolerances are fixed to 10^{-7} . Due to the high number of design variables for the K75 case, the kriging with PLS method introduced in Section 2.1 and its multi-fidelity version are used with 3 latent variables. For each test case, a gradient based optimization algorithm called Sequential Least Squares Programming (SLSQP) [18] is used to solve the enrichment optimization sub-problem from Eq. (9) with the WB2 acquisition function (see Eq. (7)). The reference value used in the K75 case, is the optimal value obtained by solving the same optimization problem with the SLSQP algorithm on the *HF* mesh. It is equals to 12.4245 and $\epsilon = 0.5\%$ in this case too. In the end, to reach the convergence criterion of the best valid objective mean value for the K75 test case, MFSEGO methodology allows to divide the SEGO total cost by 1.32.

K75 results are sum up in Tables 4 and 5 and the convergence is illustrated on Fig 3(d).

Table 3: Maximum budget and initial DoE size.

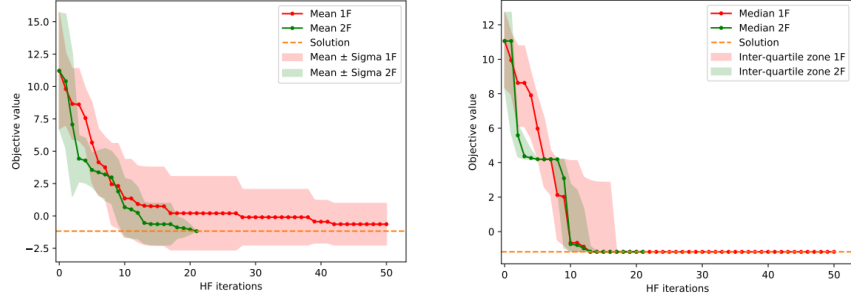
	size initial DoE (mono-fi)	size initial DoE <i>HF</i> (multi-fi)	size initial DoE <i>LF</i> (multi-fi)	budget
Branin	5	5	10	50
Sasena	5	5	10	50
K75	50	50	100	200

Table 4: *HF* evaluations needed to satisfy the convergence criterion from Eq. (17) and maximum number of *LF* evaluations over the 10 runs.

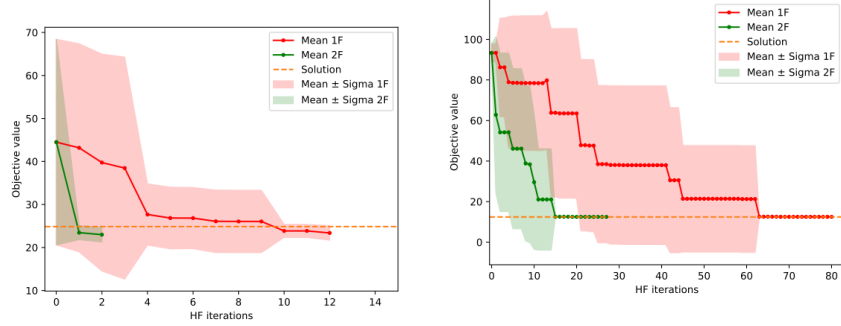
	<i>HF</i> evals (mono-fidelity)	<i>HF</i> evals (bi-fidelity)	max <i>LF</i> evals over the 10 runs (bi-fidelity)
Branin case	16	7	29
Sasena case	Don't have enough budget to converge	27	61
K75 case	131	70	144

Table 5: Total cost (one computational unit is equivalent to the cost of one *HF* evaluation) needed to satisfy the convergence criterion for each test case.

	Branin	Sasena	K75
Mono-fidelity	16	greater than 55	131
Multi-fidelity	$7 + \frac{29}{5} = 12.8$	$27 + \frac{61}{5} = 39.2$	$70 + \frac{144}{4.27} = 98.8$



(a) Mean and 1-sigma confidence interval of the objective (Sasena) value at the best valid point at each HF iteration (b) Median and interquartile range of the objective (Sasena) value at the best valid point at each HF iteration.



(c) Mean and 1-sigma confidence interval of the objective (OAS K75 model) value of the objective (Branin) value at the best valid point at each HF iteration (d) Mean and 1-sigma confidence interval of the objective (OAS K75 model) value of the objective (Branin) value at the best valid point at each HF iteration

Fig. 3: Comparison of SEGO and MFSEGO methods on the Branin, Sasena and K75 test cases.

6 Conclusion

In this work, mono and multi-fidelity Bayesian methods, SEGO and MFSEGO have been confronted. First, the two approaches were compared on analytical models. Next, a more complex test case involving aerosturctural models of the K75 drone has been considered. The MFSEGO algorithm reduces the required number of HF evaluations and the total cost needed so that the mean value (over 10 runs) of the objective function at the best valid point has a 0.5% relative accuracy. In the Branin, Sasena and K75 cases, MFSEGO allows to respectively divide the total cost by at least 1.25, 1.40 and 1.32 compared to SEGO. Future works deal with different research axis like using more than two fidelity levels, using other criteria [26] to determine the enrichment point and the enrichment level or using more complicated multi-disciplinary drone models by adding other components like operations or propulsion to the already implemented disciplines (aerodynamic and structure). As our multi-fidelity strategy can be extended to N fidelity levels, we could also study the effects of employing more than two levels.

Acknowledgements

The PhD is funded by the defense innovation agency (AID) and by the Directorate General of Armaments (DGA) as part of the CONCORDE project. This work is also supported by ONERA internal research project dedicated to multidisciplinary and multi-fidelity design optimization, namely MUFIN and is part of the activities of ONERA - ISAE - ENAC joint research group.

References

- [1] Page du d80 titan sur le site web d'elecnoir deimos, <https://elecnoir-deimos.com/project/d80-titan/>
- [2] Bartoli, N., Bouhlel, M.A., Kurek, I., Lafage, R., Lefebvre, T., Morlier, J., Priem, R., Stolz, V., Regis, R.: Improvement of efficient global optimization with application to aircraft wing design. In: 17th AIAA/ISSMO Multidisciplinary analysis and optimization conference. p. 4001 (2016)
- [3] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Bons, N., Martins, J.R., Bouhlel, M.A., Morlier, J.: An adaptive optimization strategy based on mixture of experts for wing aerodynamic design optimization. In: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. p. 4433 (2017)
- [4] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J.R., Morlier, J.: Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and technology* **90**, 85–102 (2019)
- [5] Boucher, Y., Amiez, A., Barillot, P., Chatelard, C., Coudrain, C., Déliot, P., Rivière, N., Riviere, T., Roupioz, L.: Terriscope: An optical remote sensing research platform using aircraft and uas for the characterization of continental surfaces. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2018)
- [6] Bouhlel, M.A., Bartoli, N., Otsmane, A., Morlier, J.: Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization* **53**(5), 935–952 (2016)
- [7] Bouhlel, M.A., Bartoli, N., Regis, R.G., Otsmane, A., Morlier, J.: Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Engineering Optimization* **50**(12), 2038–2053 (2018)
- [8] Bouhlel, M.A., Hwang, J.T., Bartoli, N., Lafage, R., Morlier, J., Martins, J.R.R.A.: A python surrogate modeling framework with derivatives. *Advances in Engineering Software* p. 102662 (2019). <https://doi.org/https://doi.org/10.1016/j.advengsoft.2019.03.005>
- [9] Chaudhuri, A., Jasa, J., Martins, J.R., Willcox, K.E.: Multifidelity optimization under uncertainty for a tailless aircraft. In: 2018 AIAA Non-Deterministic Approaches Conference. p. 1658 (2018)
- [10] Chauhan, S.S., Martins, J.R.: Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using openaerostruct. In: *International Conference on Engineering Optimization*. pp. 418–431. Springer (2018)
- [11] Efron, B., LePage, R.: *Introduction to bootstrap*. Wiley & Sons, New York (1992)
- [12] Forrester, A.I., Söbester, A., Keane, A.J.: Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences* **463**(2088), 3251–3269 (2007)
- [13] Frazier, P.I.: A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811* (2018)
- [14] Hernández-Lobato, J.M., Gelbart, M.A., Adams, R.P., Hoffman, M.W., Ghahramani, Z.: A general framework for constrained bayesian optimization using information-based search (2016)
- [15] Jasa, J.P., Hwang, J.T., Martins, J.R.: Open-source coupled aerostructural optimization using python. *Structural and Multidisciplinary Optimization* **57**(4), 1815–1827 (2018)
- [16] Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* **13**(4), 455–492 (1998)
- [17] Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(3), 425–464 (2001)
- [18] Kraft, D., et al.: A software package for sequential quadratic programming (1988)
- [19] Krige, D.G.: A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy* **52**(6), 119–139 (1951)
- [20] Le Gratiet, L.: Multi-fidelity Gaussian process regression for computer experiments. Ph.D. thesis, Université Paris-Diderot-Paris VII (2013)
- [21] Matheron, G., de Géostatistique Appliquée, T., Tome, I.: *Mémoires du bureau de recherche géologiques et minières*, n. 14. Ed. Technip, Paris (1962)

- [22] Ng, K.S.: A simple explanation of partial least squares. The Australian National University, Canberra (2013)
- [23] Pavlyuk, D.: Computing the maximum likelihood estimates: concentrated likelihood, em-algorithm
- [24] Priem, R.: Optimisation bayésienne sous contraintes et en grande dimension appliquée à la conception avion avant projet. Ph.D. thesis, ISAE-SUPAERO (2020)
- [25] Rasmussen, C.E., Williams, C.: Gaussian processes for machine learning, vol. 1 (2006)
- [26] Sacher, M., Le Maitre, O., Duvigneau, R., Hauville, F., Durand, M., Lothodé, C.: A non-nested infilling strategy for multifidelity based efficient global optimization. *International Journal for Uncertainty Quantification* **11**(1) (2021)
- [27] Sasena, M.J.: Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations. Ph.D. thesis, Citeseer (2002)
- [28] Watson, A.G., Barnes, R.J.: Infill sampling criteria to locate extremes. *Mathematical Geology* **27**(5), 589–608 (1995)
- [29] Wrenn, G.A.: An indirect method for numerical optimization using the Kreisselmeier-Steinhauser function, vol. 4220. National Aeronautics and Space Administration, Office of Management ... (1989)

7 Appendices

A Branin case definition

The HF and LF functions and constraints of the Branin problem are:

$$f_{Branin, HF}(x_0, x_1) = (15x_1 - \frac{5.1}{4\pi^2} * (15x_0 - 5)^2 + \frac{5}{\pi}(15x_0 - 5) - 6)^2 + 10((1 - \frac{1}{8\pi}) \cos(15x_0 - 5) + 1) + 5(15x_0 - 5) \quad (18)$$

$$f_{Branin, LF}(x_0, x_1) = f_{Branin, HF}(x_0, x_1) - \cos(0.5x_0) - x_1^3 \quad (19)$$

$$g_{Branin, HF}(x_0, x_1) = -x_0x_1 + 0.2 \leq 0 \quad (20)$$

$$g_{Branin, LF}(x_0, x_1) = -x_0x_1 - 0.7x_1 + 0.3x_0 \leq 0 \quad (21)$$

B Sasena case definition

The HF and LF functions and constraints of the Sasena problem are:

$$f_{Sasena, HF}(x_0, x_1) = 2 + 0.01(x_1 - x_0^2)^2 + (1 - x_0)^2 + 2(2 - x_1)^2 + 7 \sin(0.5x_0) \sin(0.7x_0x_1) \quad (22)$$

$$f_{Sasena, LF}(x_0, x_1) = f_{Sasena, HF}(x_0, x_1) + \exp(x_0) - x_1^2 \quad (23)$$

$$g_{Sasena, HF}(x_0, x_1) = -\sin(x_0 - x_1 - \frac{\pi}{8}) \leq 0 \quad (24)$$

$$g_{Sasena, LF}(x_0, x_1) = g_{Sasena, HF}(x_0, x_1) + 0.2x_1 - 0.7x_0 + x_0x_1 \leq 0 \quad (25)$$

C Illustration of the BO process

The six first iterations of the EGO algorithm on a one dimensional unconstrained objective function are presented in Figure 4.

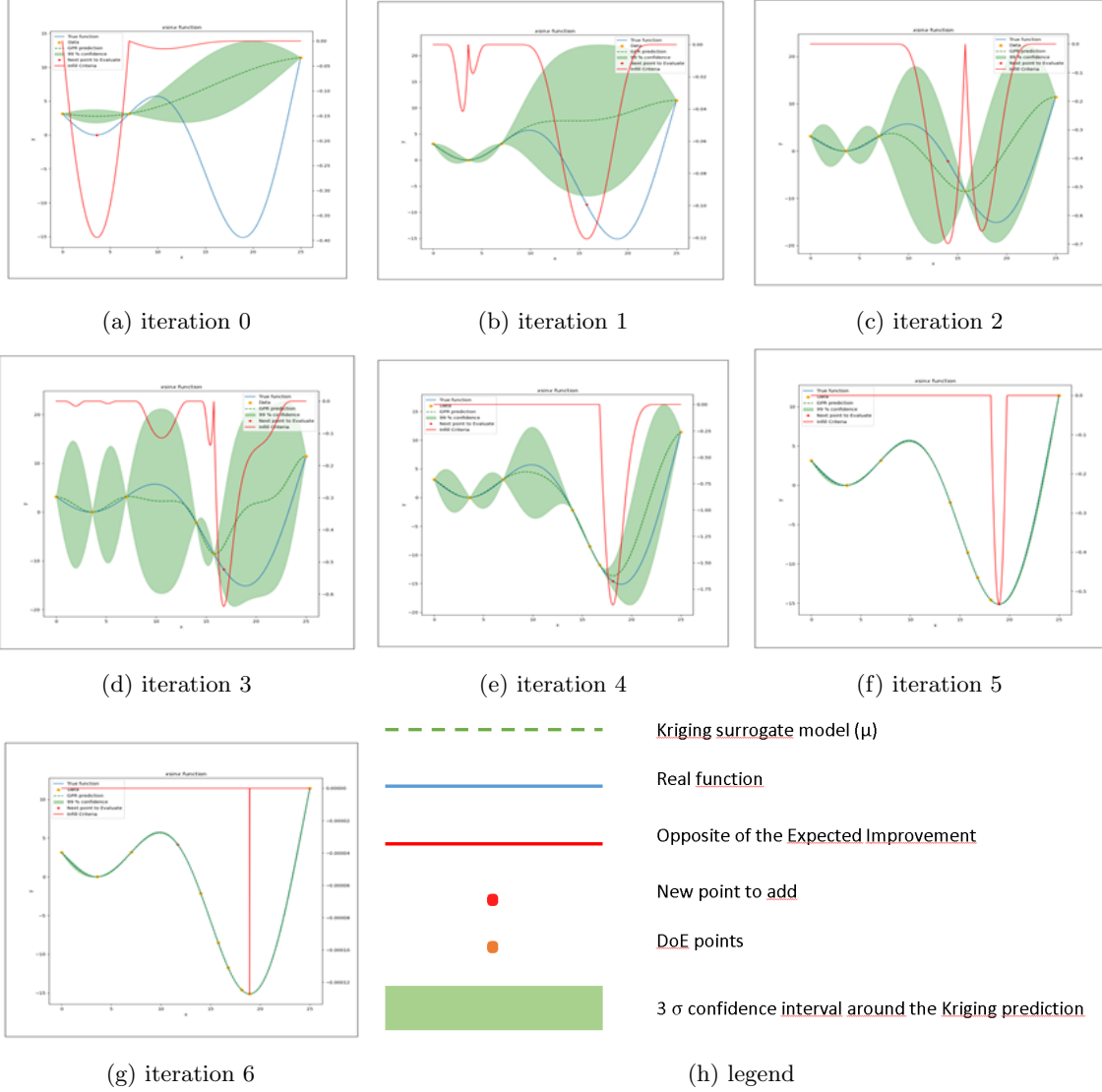


Fig. 4: Illustration of the Bayesian optimization process on a one dimensional unconstrained case: $s(x) = x \sin(3\pi(x + 0.1))$.

D K75 design variables table

Table 6: Design space and unit of the design variables

	design space	unit
3 wingbox spar thickness control points along wing span	$[0.001, 0.01]^3$	m
3 wingbox skin thickness control points along wing span	$[0.001, 0.01]^3$	m
3 wingbox spar thickness control points along tail span	$[0.001, 0.01]^3$	m
3 wingbox skin thickness control points along tail span	$[0.001, 0.01]^3$	m
angle of attack cruise flight	$[0, 15]$	deg
angle of attack maneuver	$[-15, 15]$	deg
fuelmass	$[0, 50]$	kg

E MFSEGO methodology pseudo-code

Algorithm 1 MFSEGO algorithm

```

Compute initial DoE using LHS
while (maximum budget is not reached) and ( $y_{best}^{valid} > f_{HF}(ref_{sol}) + tol$ ) do
    Learn LF Kriging surrogate model ( $\hat{\mu}_0$  and  $\hat{\sigma}_0^2$ )
    for  $k = 1 \dots L$  do
        Learn  $\rho_{k-1}$  and  $\hat{\mu}_{\delta_k}$ 
        Deduce  $\hat{\mu}_k$  and  $\hat{\sigma}_k^2$  and so the  $k$ -th fidelity level Kriging surrogate model
    end for
    Choose  $x_{next}$  that optimizes acquisition function (Eq 9)
    Select the level of enrichment  $t$  (Eq 16)
    for  $l = 0 \dots t$  do
        Add  $(x_{next}; f_l(x_{next}))$  to the DoE
    end for
     $y_{best}^{valid} = \min Y^{valid}$ 
end while
return  $y_{best}^{valid}$ 

```

F MFSEGO methodology diagram

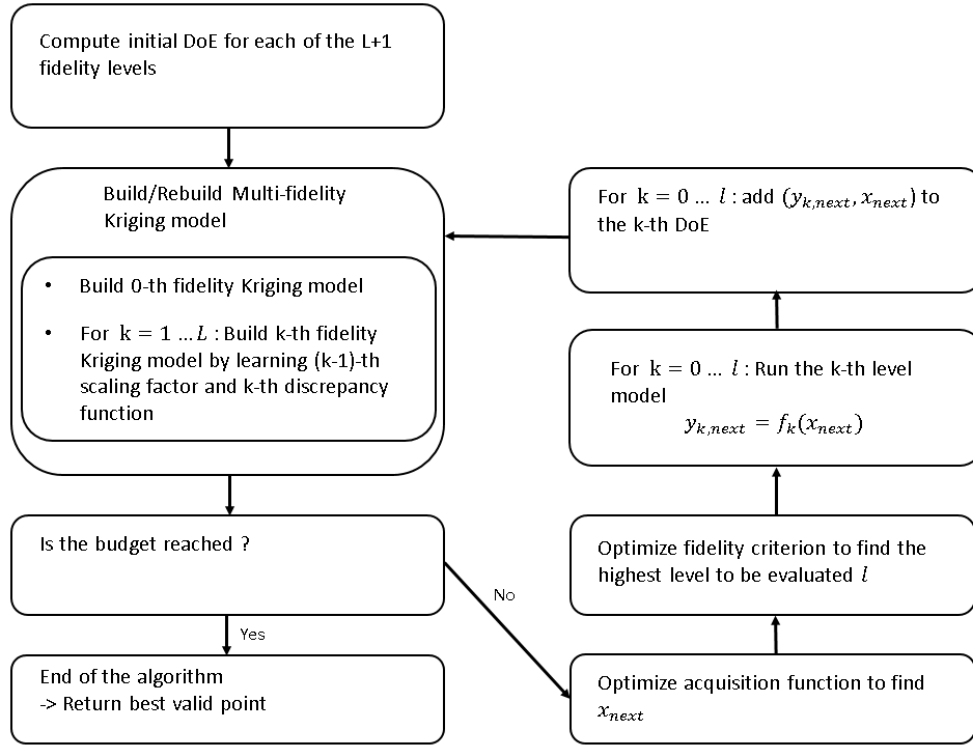


Fig. 5: MFSEGO methodology diagram

Formal Methods for AI: Lessons from the past, promises of the future*

Zakaria Chihani

CEA, LIST, Software Security and Reliability Laboratory, Palaiseau, France
`firstname.lastname@cea.fr`

Abstract. The field of Formal Methods may very well be one of the oldest fields in Computer Science, but it has been brought back to its infancy with the recent advances in Machine Learning. As more and more research teams strive to explore the safety assurance in this newly (re)discovered field, it is essential to seek insights in the history of Formal Methods, with aim of finding guidance in the current endeavour. This position paper delves into the past and offers a brief analysis of relevant similarities, in the modest hope of shedding a complementary light to the already numerous surveys.

Keywords: Formal Methods · Formal Specification · Artificial Intelligence.

1 Introduction

The roots of what is today called Formal Methods (FM) spread further in the past than their eventual acceptance and usage in the safety of software and hardware. The FM community spent decades perfecting the theories of its field, increasing its reach in industry, especially considering safety-critical systems such as transportation, energy and defense. By pushing for higher and higher standards of safety, all the while providing the tools necessary to achieve these standards, it is *no understatement* to say that the FM community made the world a safer place [15].

Unfortunately, most of the FM lore falls short in tackling the cohort of new problems brought by the recent modernisation of the old discipline that is Artificial Intelligence (AI), especially Machine Learning (ML) techniques [22]. Overcoming the opacity of the models, uncovering implicit properties and finding formalisms to specify them, detecting and repairing faulty behavior susceptible to cause significant harm, these are but a few of the large gaps in AI safety that the FM community is striving to fill. This effort is made necessary by the obvious observation that ML is shaping up to be a ubiquitous tool in our society, with permeation comparable to that of human-written software in the last decades, spanning over all aspects of our digital era. Unlike human-written code,

* This work was financially supported by European commission through the SAFAIR subproject of the project SPARTA which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892, as well as through the CPS4EU project that has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826276. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Spain, Hungary, Italy, Germany

which we will call “traditional” software in the rest of the paper, ML presents a unique set of challenges to the process of validation. While traditional software can be divided into functions that can be analysed separately, providing a structure whose semantics is usable in the context of validation and certification – each function can have its own formal specification, for example – (deep) neural networks, present no such structure and are hardly separable into smaller units. Their complexity (in terms of the different paths that an execution can have) is orders of magnitude higher than what the most mature tools can handle. In addition, while traditional software is written in more-or-less stable languages (*e.g.*, the C standard has minimal changes every decade), new ML models are invented every year (*e.g.*, new architectures and activation functions). This rapidly-changing field forces and adaptative pressure on the FM community, whose mission is to navigate around these obstacles, guided by the needs of various stakeholders (such as industrial actors, certification organisms and law-makers), to develop the tools and methods capable of ensuring a high level of confidence in AI application, helping society in harvesting the potential of this new technology without suffering from its new risks.

To find solutions to these problems, there is an increasing research effort carried out internationally, evidenced by the rapid increase in the number of workshops dedicated to this topic, such as Artificial Intelligence Safety Engineering (WAISE, at SafeComp), Safe AI (at AAAI), Machine Learning with Guarantees (ML with Guarantees, at NeurIPS), Safe Machine Learning (SafeML, at ICLR), Verification of Neural Networks (VNN, at CAV), Formal Methods for ML-Enabled Autonomous Systems (FoMLAS, at CAV), Privacy in Machine Learning (PriML, at NeurIPS), Security and Safety in Machine Learning Systems (AISecure, at ICLR), Dependable and Secure Machine Learning (DSML, at DSN), Uncertainty & Robustness in Deep Learning (UDL, at ICML).

This effervescence is encouraging, and the part of that is centered around FM is not the least developed (see, for example, applying abstract interpretation [21,19], Satisfiability Modulo Theory [13], symbolic execution [11], as well as surveys such as this one [1]). Nevertheless, there is still a persistent belief, especially in the industrial sector, that the safety of AI in general, and ML in particular, is too big a task for FM. In this position paper, we go back in time to observe the evolution of FM in relation with industrial demands, and attempt to draw a parallel between the task at hand, validation of AI, and the comparative task that required FM in the first place, the validation of human-written software. In doing so, we hope to show that the pessimism with which FM is sometimes faced in relation to AI is no more insistant than what they already faced in the past and, more importantly, that this pessimism is not necessarily justified.

Before delving in this analysis, we will now define FM and explain where they can come into play in the validation process. After that, section 2 will give a brief overview of the evolution of FM, section 3 explains the specification problem, section 4 discusses the scalability issues, and we conclude with a brief summary of the lessons that can be learnt from this history.

1.1 Formal Methods

In its most strict definition, a formal method is intrinsically linked with proof, and therefore only concerns tools such as theorem provers and proof assistants. What is not included in this definition is any testing methodology, which, as Dijkstra famously said,

can help you find bugs but not prove their absence. However, in the broader sense that we consider here, a formal method is defined by two criteria: (1) it is deployed for the validation of software or hardware; (2) it relies on mathematical and logical background.

This definition still excludes testing a software through the use of a simulator, be it traditional or AI-based (*e.g.*, Generative Adversarial Network, or GAN). Indeed, an image generating engine (not unlike those used in video games, pilot training, *etc.*) capable of simulating a certain scenario (*e.g.*, car accident, to see how an autonomous vehicle would behave) does not have at its source a logical description of specifications relevant to a given domain of application. In contrast, property-based testing *is* a formal method: it relies on logically and mathematically specified properties to generate tests.

In reality, FM are also used for the specification, development and monitoring of software. The FM discipline then is articulated around three pillars: the methods and tools themselves (for analysis, test, verification *etc.*), the properties to ensure (*e.g.*, through a global formal proof, through a local assurance with property-based testing, through runtime monitoring during the deployment), and finally the object itself, whose life cycle should be covered as much as possible through the rigorous the usage of FM. With regards to the AI-enabled objects, we would like to make two observations:

AI-specific challenge: the practitioners of FM usually enjoy a certain stability. For example, some tools are specialized in the analysis of C code, and have a certain number of industrial actors that use them to verify their programs. If a new, better language (*w.r.t.* to some metric, readability, garbage collection, *etc.*) appears, the industrial actor would need to invest a considerable amount of time and money to recode the potentially million-lines-long computer infrastructure. For this reason, tools that analyse C, whose standard only evolves every decade or so, are a relatively safe bet. For ML, however, it is another story. Not only new activation functions appear on the regular, but even new architectures (*e.g.*, Graph NN, Capsule networks), and dealing with them requires a much more adaptation effort than for a C analyser.

FM-specific AI-choices: The history of software development gives many examples where the validation process influences the programmers' behaviour and choices. One only has to remember the struggle [16] that was needed in order to make programmers abandon the "goto" statements, which were very convenient yet catastrophic for any validation process. A similar dynamic needs to take place between FM and AI practitioners. *e.g.*, an ML practitioner can be encouraged to use some activation functions rather than others if the loss in accuracy is not dramatic and if the gain in the validation process is significant.

1.2 Trust issues

Several properties are essential to increase the trust in a deployed software. Privacy, fairness, safety and reliability (ability of a system to perform its functions in nominal conditions), security (behaviour of a system under an attack by a malicious agent), explainability and interpretability (different levels of intelligibility of the software), *etc.* As said above, FM rely on formal specification (logically and mathematically described properties) as a basic pillar. Because of this, some trust issues are better suited than others to the usage of FM. These are usually safety and security concerns. Indeed, even if

some fairness questions can be formally specifiable, ethical questions are usually hard to formalize. This is not a failure of FM, as they *can* be used if and when there is a formally specified property. *e.g.*, if one wants to verify the fairness of a CV-analysing program P *w.r.t.* origin, which we can describe by “for all other inputs (*e.g.*, education e , previous jobs j , age a) from two CV that are within a certain distance ε of each other, and different origins o_1 and o_2 of two candidates, the two assessments should be within a certain distance of each other”. Formally, $\forall i \in \{1, 2\}, e_i, j_i, a_i, o_i. \|e_1, e_2\| < \varepsilon_e, \|j_1, j_2\| < \varepsilon_j, \|a_1, a_2\| < \varepsilon_a \Rightarrow \|P(e_1, j_1, a_1, o_1), P(e_2, j_2, a_2, o_2)\| < \varepsilon_P$. (of course, one has to also detail the relationship between the different ε values). However, as said above, this remains the minority of the usage of FM. We will therefore mainly discuss the type of properties that FM are most commonly used for and are starting to be used for AI in general and NN in particular.

Robustness properties. One indicator of the reliability of a system is its stability or robustness to input perturbations. Specifically, robustness requires that all samples in the vicinity (some distance metric) of a given input are classified with the same label. Robustness is both a safety and security property, the difference is mainly related to the scenario of the perturbation. If it is the result of a *targeted* attack by a thinking agent, that adapts to the model’s vulnerabilities [10], the issue is one of security. The difficulty of this side of software trust is that it is virtually impossible to protect a system against *all* possible attacks. This domain is usually a never-ending arms race between the defenders and the attackers. If the perturbation is the result of natural setting, it often enjoys properties that can be taken into account. For example, certain lighting condition influencing the visibility, or certain noise troubling an audio input, could be described, in some cases, as a familiar shape such as a Gaussian.

Metamorphic properties. These properties [2] do not describe the relationship between inputs and outputs, but between relationships of inputs and relationships of outputs, and is often used in property-based testing (even if formal verification is also possible). Consider the most common example: a software S that computes the minimal cost of travel between two points, a and b , in an undirected graph. Even if the actual result of this operation is not known, it is possible to generate an arbitrary number of tests using the knowledge that the result should be impervious to symmetry. Here, the relation between the inputs (a, b) and (b, a) is the symmetry, and the relationship on the outputs $S(a, b)$ and $S(b, a)$ is the equality. But the relationships can be more complex. Consider an evolution S' of the software S above, that takes as input a point of origin o and a list L of points and computes the lowest cost for package delivery to all of them and come back to the origin. One can generate numerous partitionnings of L into lists L_1, L_2, \dots , and the sum of the results of calls $S'(o, L_i)$ should be greater or equal than the result of $S'(o, L)$. In other words, $\forall L_1, L_n, L = \bigoplus_{i=1}^n L_i \rightarrow S'(o, L) \leq \sum_{i=1}^n S'(o, L_i)$, where \bigoplus is the list concatenation.

Semantic properties. These properties require a deep understanding of the application domain and specify the behaviour of the system in its globality. They are usually the result of the decomposition of higher-level specification into smaller properties, formalized in order to be readily treatable by computer tools. These can be related to the inner

workings of the system (*e.g.*, a certain memory zone is never accessed) or to the possible outputs (*e.g.*, the autopilot will never issue a nose-dive directive).

2 Failed prophecies

“Program verification is bound to fail. We can’t see how it’s going to be able to affect anyone’s confidence about programs”.

The above quote is from an influential 1979 paper [17], and it is barely out of context. The authors found many justifications for their claim and, to their defense, this was not an isolated occurrence but came at a time where there were quite strong debates on the subject. But it is now a few decades later and we can be reasonably, objectively confident that they were wrong. FM are routinely used today in the validation process of several (mostly safety-critical) domains such as aviation, energy and defense.

The reader might think that these were misguided unexperienced researchers. But that is not the case: Richard De Millo is a distinguished Professor of Computing at the Georgia Tech and served as VP and CTO of Hewlett-Packard. Richard Lipton taught at Yale, Berkeley, Princeton, Georgia Tech, he’s also a fellow at the ACM and a Knuth Prize winner. Alan Perlis was a professor at Purdue, Carnegie Mellon and Yale, and is the first recipient of the Turing Award.

Unfortunately, their voice, along with likeminded others, was not unsequential. Debates between great minds rarely are. This critique had a direct influence on the development of FM in mainly two ways. The first is social, weighting on what courses were taught at university, what choices of PhD subjects students would make. The second is financial, reducing funding opportunities and impacting even the language used. Like the words “artificial intelligence” during the AI winters, the words “program verification” became somewhat toxic and could hurt the credibility of submitted projects.

This negative perception changed overtime and one pivotal moment was no doubt the crash of Ariane 5, both because it could have been prevented had certain FM been used, and because it is historically the first time abstract interpretation [4] (one of the most successful FM) has been used in the validation of a project of this importance. In the following years, FM continued to gain in importance, even reaching official recognition when some certification standards started calling for (*e.g.*, DO-178C and its DO-333 supplement) or even mandating (*e.g.*, ISO/IEC 15408) their use. More recently, FM have been adopted in less critical environments, *e.g.*, by Facebook [5] and Amazon [3], which shows the level of maturity and scalability of these methods. The full history of the mechanization of proofs is laid out expertly in a book [16], whose pedagogical style and lack of technical jargon make it a very agreeable read even to the non-specialist (the author, Donald MacKenzie, being a sociologist and historian).

Before closing this section, it should be pointed that the context above was given not to disparage researchers who more than earned their places in the history of Computer Science, but to insist on the fact that even great minds can be wrong. And when the possible payoff is a safer auto-pilot for a plane or a space-shuttle, or a more robust software for the cooling systems of nuclear power plants, perhaps absolute opinions should be taken with a grain of salt.

3 The specification problem

As said above, one of the pillars of the FM discipline is the formalized specification of the desired properties that one would like to ensure in one's software. In here lies a significant challenge to FM when it comes to AI, especially its ML sub-field. One can distinguish two possibilities:

Direct specification: The AI can be a surrogate for a programmable task, which often happens in the case of some control and command software that relies on quite sizeable reference tables where every possible input combination is associated with some directive. For example, in Air-Collision Avoidance Systems, or ACAS, the inputs can be the speed, angle of approach and the distance, and the output is some avoidance directive (typically to make a turn). In this case, each individual input has a *semantic value* that can be used in a formal specification. One can, for example, verify that if the speed is between some bounds, and the distance is between other bounds, then the directive is necessarily a left turn directive. This best case scenario is not rare, but it is unfortunately not where the recent advances of ML brought the real revolution.

Indirect specification: The AI can approximate a function that cannot be written as an algorithm. This is the main reason behind the renewed interest in ML. In fact, it is through state-of-the-art performance in an image recognition competition that the new potential of AI was made apparent about a decade ago. But from the point of view of FM, this brings a significant challenge: how to *formally* specify what a car is? Here, the individual inputs, (in the case of images, the pixels) do not have an intrinsic semantic value. The phenomenon (*e.g.*, a car, a pedestrian, a song) that one would like to describe in a formal specification is an *emergent concept* from small parts (pixels, wavelengths) that do not individually exhibit that concept. A concept such as speed, distance, altitude, which is immediately described by its numerical value, can directly appear in a formal specification. But for concepts such as images, the specification has to contend with inputs that are abstract components of the concept, not the concept itself.

In the latter case, the formal specifications make use of general descriptions such as the neighbourhood of an image in terms of the possible variations on its individual pixels, or the distance between two images as a cumulative difference between their respective pixels. But the emergent concept (*i.e.*, what the image actually contains) cannot immediately appear in the specification. This prompted several avenues of research into what can serve as a specification, and that we will now discuss.

3.1 New definitions of specification

“Do the best you can until you know better. Then when you know better, do better.”

– Maya Angelo

One could argue that the goal of FM in particular and certification in general is to reach the highest *possible* standard of trust in our software, and that is often subject to some restrictions and overapproximations. Before it is released on the roads, a car is required to run a field test of a certain amount of hours. Each plane crash is carefully

analysed and lessons are learnt from tragedies in order to reduce future risks. When faced with situations that are not 100% predictable, we tend to carve a subset of the possible inputs that *can* be described and we make a safety case out of that. The notion of “acceptably safe” is, indeed, a common one in certification circles.

Surrogate model in the specification: in this case, one could assess the adequacy of an efficient AI *w.r.t.* a non-efficient, non-AI, but well understood surrogate model.

Simulator in the specification: one can devise a setting [9] where the subset of inputs is delimited by the possible outputs of a simulator. In this case, either the simulator itself is formalised (for example, through the usage of a GAN [7] or the simulator outputs some kind of certificate for what it simulates). If the simulator is trusted, (arguably a big “if”), then it could be used in the specification.

Dataset in the specification: currently, a neural network is trained on a dataset then tested and validated on others. Therefore, the community has already chosen to trust some data enough to both train our models and validate their adequacy. If that is an acceptable step, it stands to reason that using a validation dataset, *e.g.*, to prove robustness *w.r.t.* perturbations around each of the data points in that set, should be at least no less acceptable to achieve the same, if not stronger, trust.

Some other guarantees, such as probabilistic approaches [24] are possible, but it may be the case that we will never reach the level of trust that we currently have in, say, plane autopilots. The acceptable level of trust for these systems will require much longer debates that have already started at many major certification and standardisation agency in the world (*e.g.*, ISO, NIST, Afnor, LNE, Bureau Veritas).

4 Scalability

A common criticism raised against the usage of Formal Methods for AI is that of scalability. In a domain where the preponderant technique, *i.e.*, Machine Learning, often produces objects (mostly neural networks) that contain millions of hyper-parameters, raising doubts on the tractability is a natural reflex, and an arguably valid one, considering the combinatorial explosion that ensues from the analysis of such artefacts. However, here again, it can be informative to take a look at the evolution of FM in the past, which we will now do by taking SAT-solving as an example.

Boolean satisfiability, or SAT, is the name of the problem of finding values for the variables of a Boolean formula in order to make it true. SAT is the first known NP-complete problem (not only is it NP-hard, but it can also encode any NP-hard problem). Because it is NP-complete, all algorithms known for SAT solving have an exponential worst-case complexity. This did not stop the FM community from developing efficient and scalable algorithms that can automatically solve problems with tens of thousands of variables and millions of constraints. Indeed, the first SAT-solving algorithms could not scale to large problems, it took decades of research into efficient algorithms and clever heuristics (DPLL, CDCL, Symmetry breaking, two-watched literals, WalkSAT, adaptive branching, random restarts, portfolio methods, divide-and-conquer, parallel local search, *etc.*) to achieve the current state of the art.

It is sometimes through the analysis of some problems that heuristics are developed (which makes them more efficient for their targetted problems but sometimes less effective in general). Such an analysis can take place in the field of validation of ML, with the definition of new abstract domains to overapproximate a NN behaviour, or new heuristics that are based on the activations of individual neurons to guide the search.

Far from being discouraged by the complexity of neural networks, more and more members of the FM community are rising up to the challenge and investigating this still largely unexplored field. And already, more than 20 solvers (*e.g.*, MaxSens [25], Duality [6], DLV [12], Certify [18], AI² [8] and ERAN [20], ReluVal [23], Marabou [14]) have emerged, either through adaptation of previous tools or through the development of dedicated ones. And year after year, new improvements and adjustments push their precision and scalability to very encouraging levels. The position that FM will never be able to scale to large problems does not seem to be a justified prophecy.

5 Conclusion

The FM community is familiar with the current state of affairs of the research in the field of trustworthy AI, having gone through a similar evolution in the past for traditional, human-written software. Crucially, the current evolution is much faster:

- Cambrian explosion: Just in the past few years, more than 20 tools were released.
- Competition for resources: Each paper published increases the scalability.
- Cross-fertilization: Good ideas from one tool are implemented in others.
- Niche creation: Some solvers specialize into particular models and type of properties.
- Adaptative Pressure: New models, new architectures, and in general new AI-technologies are born every year and the tools to validate them must keep up.
- Domestication: ML practitioners should be made aware of the choices in implementation that can make their models more amenable to FM, so that they can factor this aspect in their decision process.

We hope to have convinced the reader that, by looking at the past and the present, one can only reach the conclusion that FM still have an important role to play in the trustworthiness of AI.

References

1. Bunel, R.R., Turkaslan, I., Torr, P., Kohli, P., Mudigonda, P.K.: A unified view of piecewise linear neural network verification. In: Advances in Neural Information Processing Systems. pp. 4790–4799 (2018)
2. Chen, T.Y., Kuo, F.C., Liu, H., Poon, P.L., Towey, D., Tse, T., Zhou, Z.Q.: Metamorphic testing: A review of challenges and opportunities. ACM Computing Surveys (CSUR) **51**(1), 1–27 (2018)
3. Cook, B.: Automated formal reasoning about amazon web services (keynote). In: Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software. p. 9. SPIN 2017, Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3092282.3092315>, <https://doi.org/10.1145/3092282.3092315>

4. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: POPL. pp. 238–252 (1977)
5. Distefano, D., Fähndrich, M., Logozzo, F., O’Hearn, P.W.: Scaling static analyses at facebook. *Commun. ACM* **62**(8), 62–70 (Jul 2019). <https://doi.org/10.1145/3338112>, <https://doi.org/10.1145/3338112>
6. Dvijotham, K., Stanforth, R., Gowal, S., Mann, T.A., Kohli, P.: A Dual Approach to Scalable Verification of Deep Networks. In: Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6–10, 2018. pp. 550–559 (2018), <http://auai.org/uai2018/proceedings/papers/204.pdf>
7. Fijalkow, N., Gupta, M.K.: Verification of neural networks: Specifying global robustness using generative models (Oct 2019)
8. Gehr, T., Mirman, M., Drachler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: AI2: Safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE Symposium on Security and Privacy (SP). IEEE (may 2018). <https://doi.org/10.1109/sp.2018.00058>
9. Girard-Satabin, J., Charpiat, G., Chihani, Z., Schoenauer, M.: Camus: A framework to build formal specifications for deep perception systems using simulators. ECAI (2020)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples (2015)
11. Gopinath, D., S. Pasareanu, C., Wang, K., Zhang, M., Khurshid, S.: Symbolic execution for attribution and attack synthesis in neural networks. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 282–283 (May 2019). <https://doi.org/10.1109/ICSE-Companion.2019.00115>
12. Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: Computer Aided Verification, pp. 3–29. Springer International Publishing (2017)
13. Katz, G., Barrett, C., Dill, D., Julian, K., Kochenderfer, M.: Reluplex: An efficient smt solver for verifying deep neural networks. arXiv preprint arXiv:1702.01135 (2017)
14. Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D.L., Kochenderfer, M.J., Barrett, C.: The Marabou Framework for Verification and Analysis of Deep Neural Networks. In: Dillig, I., Tasiran, S. (eds.) Computer Aided Verification, vol. 11561, pp. 443–452. Springer International Publishing, Cham (2019)
15. Klein, G., Andronick, J., Fernandez, M., Kuz, I., Murray, T., Heiser, G.: Formally verified software in the real world. *Communications of the ACM* **61**(10), 68–77 (2018)
16. MacKenzie, D.: Mechanizing Proof. MIT Press (2001)
17. Millo, R.A.D., Lipton, R.J., Perlis, A.J.: Social processes and proofs of theorems and programs. pp. 271–280 (May 1979)
18. Raghunathan, A., Steinhardt, J., Liang, P.: Certified defenses against adversarial examples. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), <https://openreview.net/forum?id=Bys4ob-Rb>
19. Ranzato, F., Zanella, M.: Robustness verification of support vector machines. In: International Static Analysis Symposium. pp. 271–295. Springer (2019)
20. Singh, G., Gehr, T., Mirman, M., Püschel, M., Vechev, M.: Fast and Effective Robustness Certification. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 10802–10813. Curran Associates, Inc. (2018), <http://papers.nips.cc/paper/8278-fast-and-effective-robustness-certification.pdf>
21. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages* **3**(POPL), 1–30 (2019)
22. Vassev, E.: Safe artificial intelligence and formal methods. In: International Symposium on Leveraging Applications of Formal Methods. pp. 704–713. Springer (2016)

23. Wang, S., Pei, K., Whitehouse, J., Yang, J., Jana, S.: Efficient Formal Safety Analysis of Neural Networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 31, pp. 6367–6377. Curran Associates, Inc. (2018), <http://papers.nips.cc/paper/7873-efficient-formal-safety-analysis-of-neural-networks.pdf>
24. Weng, L., Chen, P.Y., Nguyen, L., Squillante, M., Boopathy, A., Oseledets, I., Daniel, L.: Proven: Verifying robustness of neural networks with a probabilistic approach. In: *International Conference on Machine Learning*. pp. 6727–6736. PMLR (2019)
25. Xiang, W., Tran, H.D., Johnson, T.T.: Reachable set computation and safety verification for neural networks with relu activations (Dec 2017)

The benefits of using ALTAI in Defence applications

Bruno Carron and Stéphan Brunessaux

Airbus Defence and Space, Elancourt, France

`bruno.carron@airbus.com`, `stephan@brunessaux.com`

Abstract. Controversial discussions about the use of Artificial Intelligence have raised public attention in critical domains including automobile, health, and defence with questions about the responsibility in case of failure of AI-based systems. This has led to important ethical and legal implications that kicked off different studies across the world. In Europe, the works of the High-Level Expert Group on Artificial Intelligence produced different reports for achieving trustworthy AI including the Assessment List for Trustworthy Artificial Intelligence (ALTAI). Though not specifically designed for military AI-based systems, ALTAI provides a list of questions to self-assess the trustworthiness of an AI-based system. Our study has consisted in applying ALTAI on a military use case - Target Detection, Recognition and Identification - in order to assess the applicability, benefits and/or limitations of ALTAI. Our work is conclusive about the benefits of ALTAI and the questions listed in ALTAI has helped with the identification of many aspects to improve the trustworthiness of our AI-based system. With this paper, we want to encourage all defence-AI based system developers and stakeholders to use ALTAI in order to achieve AI trustworthiness.

Keywords: Trustworthy AI, Ethical AI, Responsible AI, AI for Defence, Assessment, ALTAI, Transparency, Technical Robustness, Safety, DRI

1 Introduction

There is very little doubt about the significant impact of Artificial Intelligence [1] on just about every aspect of daily life. Controversial discussions, such as in the automotive industry in the context of autonomous driving, have raised public attention which eventually also reached the military domain with questions about ethical and legal implications of the military usage of AI [2].

In France [3], DGA released early 2021 a second version of its methodological guide for the specification and qualification of AI-based systems [4] while in the US, after recommendations from the Defence Innovation Board [5], the Department of Defence adopted in 2020 AI Ethical Principles for the design, development, deployment, and use of AI capabilities for both combat and non-combat purposes [6]. These initiatives and others were conducted while the High-Level Expert Group on Artificial Intelligence set up by the European Commission was designing a self-assessment list for trustworthy AI-based systems.

The objective of this paper is to report about and share the results of an internal study [7] started last year and still on-going that ultimately consists in designing an approach to develop responsible AI-based military systems addressing ethical and legal questions that may arise when using such technologies.

The paper is organised as follows. Section 2 presents the work of the High-Level Expert Group on Artificial Intelligence on Trustworthy AI. It is considered as one of the, if not the most, comprehensive and exhaustive work on the development of trustworthy AI-based system thanks to a set of practical guidelines/questions to address. Section 3 presents the use case that was selected to experiment with the ethics guidelines. Section 4 illustrates the application of ALTAI on the use case and the recommendations that were derived from the work. Section 5 gives the key takeaways of this study and section 6 provides a conclusion. All relevant references are compiled at the end of this article.

2 Ethics Guidelines on Artificial Intelligence

In June 2018, European Commission's Directorate-General for Communications Networks, Content and Technology (DG CNECT) set up the High-Level Expert Group on Artificial Intelligence (AI HLEG) with the mandate to define Ethics Guidelines for Trustworthy AI. This independent group was composed of 52 appointed experts from academia, small and medium enterprises (SME), NGOs, and large groups.

A first report [8] called “Ethics Guidelines on Artificial Intelligence” was released in April 2019. These Ethics Guidelines are organised around four ethical principles based on fundamental rights:

1. **Respect for Human Autonomy:** Human dignity encompasses the idea that every human being possesses an intrinsic worth, which should never be diminished, compromised or repressed by others – nor by new technologies like AI systems.
2. **Prevention of Harm:** AI systems should neither cause nor exacerbate harm or otherwise adversely affect human beings.
3. **Fairness:** The development, deployment and use of AI systems must be fair.
4. **Explicability:** Processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected.

An initial assessment list comprising 131 questions has been proposed as a practical tool to check that the AI system being developed could be qualified as trustworthy. This initial report was complemented with another document [9] called the “Assessment List for Trustworthy Artificial Intelligence (ALTAI)” (for self-assessment) released in July 2020 together with a Web-based tool.

The ALTAI questions, which are a reformulation of the initial 131 questions, are clustered according to the 7 key requirements¹ as depicted in figure 1.

¹ The interested reader is invited to refer to the Ethics Guidelines [8] to get detailed information about these requirements.

For each of these requirements comes a list of questions that varies in number and complexity. It is recommended to take time and brainstorm with colleagues to ensure a proper understanding of the questions and identify their rationale. ALTAI is accessible on-line and off-line and the authors have preferred to go through the 34-page document. It has proved to be more suited to an in-depth analysis of the questions and an assessment of their applicability to military AI-based use cases. Measures to take (if any) have been recorded to ensure that the AI-based system could be considered as trustworthy if such recommendations were to be followed.

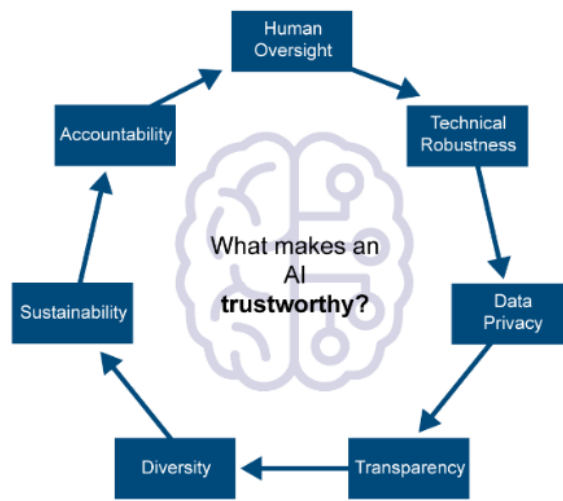


Fig. 1. What makes an AI Trustworthy (based on [8])

To give an idea, here are examples of ALTAI questions that one has to reflect on when self-assessing the trustworthiness on an AI-based system:

- *Did you put measures in place to ensure the integrity, robustness and overall security of the AI system against potential attacks over its lifecycle?*
- *Can you trace back which data was used by the AI system to make a certain decision(s) or recommendation(s)?*
- *Did you identify the possible threats to the AI system (design faults, technical faults, environmental threats) and the possible consequences?*
- *Did you put in place measures to continuously assess the quality of the input data to the AI system?*
- *Do you continuously survey the users if they understand the decision(s) of the AI system?*
- *Did you provide appropriate training material and disclaimers to users on how to adequately use the AI system?*
- *Did you communicate the benefits of the AI system to users?*

ALTAI has been applied to the DRI use case that is presented hereafter.

3 Target Detection, Recognition, and Identification

The use case Target Detection, Recognition, and Identification (DRI) [10] covers the use of AI technology to detect and identify potential targets as part of the *Find* and *Fix* activities of the F²T²EA cycle (Find/Fix/Track/Target/Engage/Assess). It includes the processing of imagery and video data, in the military domain commonly referred to as Automatic Target Recognition (ATR), as well as the analysis of RF signal intercepts, also known as Cognitive Electronic Warfare (EW). In these methods, supervised learning is most commonly used to detect pre-defined patterns within the analysed signals.

The operational benefit of AI-based DRI is the huge amount of data, which can be processed in short time. Modern computers can perform this task in real-time and in a massively parallelised fashion without fatigue. Thus AI-based DRI in support of a human operator is finally able to accelerate the *Find* and *Fix* activities during the F²T²EA-Cycle, which is highly critical for time-sensitive missions. As of today, the underlying methods are sufficiently mature to be deployed in operational systems for standard operational conditions and thus have become more or less state-of-the-art in the last couple of years. Based on that, performance improvements for DRI are expected within the next decade for adverse environmental conditions and difficult low observable targets.

DRI itself can be regarded as a highly automated use case, as all processing steps will be fully delegated to the machine. The only input given by the operator might be a directive of what data to analyse and a schema describing the features of interest for the scope of the mission.

A priori, the ethical concerns induced by the use of an AI-based DRI depend on the system characteristics:

- Supervised system: In case the interpretation of the generated results is regarded to be a task for a human operator again, AI-based DRI is not seen critical from an ethical perspective, as human judgement will be superordinate in the process.
- Fully automated system: However, in case a system will be designed to automatically take engagement decisions based on DRI results, it will certainly become the most critical ethical aspect for the use of AI in weapon systems.

4 Assessment of the DRI use case

The list of questions proposed by ALTAI support the assessment of the trustworthiness of an AI-based system, independently of the level of maturity of the system. In our study, the analysis was conducted at a very early stage of our DRI AI based system: TRL 2 [11]. As a result, this trustworthiness analysis work could lead to very specific recommendations for the specification, design or implementation phases as well as during the operational use of the DRI system.

During this analysis, DRI was considered as a non-critical system due to an operator in the loop, as opposite to an automatic DRI system that would be fully automatic. The

next sections illustrate the work carried in this study by giving an overview of the recommendations that were produced on 2 of the 7 requirements. This clearly demonstrates the real added value of ALTAI to develop a trustworthy AI-based system.

4.1 Technical Robustness and Safety

The ALTAI questions regarding this top-level requirement are organised into 3 sub-categories: Resilience to Attack and Security, General Safety, and Accuracy. The questions, which are too numerous to be detailed in this paper, helped with the identification of various recommendations, which are summarized hereafter.

Regarding **Attack and Security**, if current practice of DRI is considered compliant with security standards for military networks and equipment, it was acknowledged that the use of AI could bring opportunities for attackers. Therefore, our recommendation has been to refer to the work of the ad-hoc expert group of the European Union Agency for Cybersecurity that released a report [12] that helps identifying all the cybersecurity threats that an AI system could face. The implementation of a Swiss cheese model has also been recommended as it combines technology, process and people to reduce vulnerabilities. Among the possible threats, the compromise of a human developer of an AI component was flagged for particular attention. This could be solved by setting up specific organizational processes.

Regarding **Safety**, we have recommended to use the user interface to give a transparent information to the DRI operator about the level of confidence the AI component has on D, R and I, to display the reference data, and to trace the evolution performed after the factory qualification. When identifying possible threats and counter-measures, we have suggested to consider the system at its 3 distinctive levels: the AI algorithm, the AI based system and the human operator interacting with the AI system. In the interface between the AI based system and the operator, we have recommended to put the choice in the hand of the operator, without any default behaviour, and to be transparent on the possible failures.

Regarding **Accuracy**, our recommendation has been to conduct a deep dive analysis of the consequence of the potential errors of the AI based system, and ensure its understanding by the operator. Another suggestion has been to set boundaries based on the risk based analysis and to forbid the use of the AI system when it is out of the boundaries or not measurable. Additional recommendations have been to set up a data management plan (including logbook of the data, including dates and access control list to check operator clearance) and consider the use of frugal technology. It has also seemed important to continuously record and monitor the detailed performance of the system, with the computation of standard AI errors measurements such as precision and recall [13], and determine what the acceptable error rates are. To qualify the performance of the system, and ensure it is used in the proper operational conditions, it is also important to specify the AI based system and not solely the AI algorithm.

4.2 Transparency

The ALTAI questions regarding this top-level requirement are organised into 3 sub-categories: Traceability, Explainability, and Communication. The recommendations derived from applying ALTAI are summarized hereafter.

To achieve **Traceability** in DRI, we have recommended to continuously assess the training data and the operational data, including when they are Governmental Furnished Equipment. Logging mechanisms have to be implemented to trace between decisions of the AI-based system and the data triggering the decisions. A warning has been issued on a possible storage issue (size of video files to log). Attention has also been drawn on the need to implement confidentiality mechanisms when potentially logging classified information. The development of a confidence technical indicator has also been proposed to achieve better understanding of the decision.

To achieve **Explainability**, we have recommended that the DRI system includes this functionality either as an integral part of the system (explainable-by-design) or as a side function. Explanations have to be made available to the DRI user in real time. We have also suggested to record the course of actions with the decision of the AI system. The presentation of the DRI outputs to the operator has to follow ergonomics recommendations, with the ability to provide feedback to the system. In addition, it has been recommended to conduct regular training/satisfaction survey with the DRI operators.

To achieve better **Communication** and transparency, it is has again been recommended to record logs on the AI system and to compute some statistics on DRI activity and performances: number of detections, R, I, etc. These logs should be made available to the operator. The analysis of this information will be key for improving the AI system, and identify any potential failure related to sensors or other aspects. Training is also an essential element for increasing transparency. It should be adapted due to the use of AI technologies. The technical limitations and potential risks induced by the use of AI should be communicated to the operators while still mentioning the rationale and benefits of using AI in the context of DRI.

5 Key takeaways

By applying the ALTAI methodology to our use case, we have observed multiple consequences that we can generalize for the Systems Engineering of defence AI systems:

1. ALTAI helps defining additional requirements, applicable to various steps of the system lifecycle: Design, Development, Operations.
2. AI based systems should be considered according to systems engineering standards, and not only the AI algorithm as an isolated component. With this approach, the operator becomes a part of the system to care about, as well as any additional features, for instance checking the validity of the input/output values of the defence system.

3. Items 1 and 2 above enables risk analysis, which is a Systems Engineering practice for critical systems. Levels of criticality and systems failure modes need to be defined, which takes into account how the systems will be used.
4. The whole system lifecycle is impacted: design, development, delivery, operation, dismantling. This is needed considering the companies' ethical choices as well as the European and National statements/regulations on Artificial Intelligence.

6 Conclusion

This paper has addressed the concerns of a responsible use of AI in defence applications that is more vivid than ever with the current ethical debate about the use of AI in risky applications as recently emphasized by the European Parliament with its proposed Artificial Intelligence Act [14] to regulate the use of Artificial Intelligence.

The study reported here has shown that applying ALTAI developed by the AI HLEG was highly valuable though not always straightforward and often time-consuming (as for all important ethical questions). It usually takes some time to understand the rationale and the subtleties of the various questions before being able to imagine ways of addressing the ethical challenges raised by these questions and the potential consequences. Despite the fact that ALTAI was not particularly designed to assess trustworthy AI-based military systems, the study demonstrated that it is fully applicable and that the questions are always relevant and helpful in the development of a trustworthy AI-based system and the authors strongly suggest all developers of AI-based systems, risky or not, to use ALTAI and properly address all the questions it raises. Trustworthiness can be achieved either by making specific design changes to the system or by setting up additional processes to ensure a responsible use of the system. One important lesson was to consider the system as a whole and not solely focus on the AI component(s).

The question whether ALTAI in its current version should be extended or tailored for defence applications in general, does require a more detailed analysis and will be discussed further, given the potential lethal consequences of military decision making that cannot just be attributed to performance measures of a technical system. Among the specificities of AI-based defence applications are the need for increasing speed of the course of action and the search for information superiority while being used in a hostile or denied environment.

Numerous additional points remain open such as the reduction of potential biases in algorithms, the potential negative impact of the acceleration of decision-making to the escalation of a combat situation, the lack of immunity of algorithms to traditional means of espionage and deception, the risk of an over-sized degree of authority in political decision-making of AI-generated analyses.

However, there is still the opportunity to go for a European way that keeps the overall system under control of an informed, aware, and accountable human operator, which is equipped with means of control that are meaningful to the required and spec-

ified level. To do so, the dialogue of experts and advisors from across academia, military domain, politics, and civil society is to be continued. Based on insights gained therein as well as on cross-domain collaborative research on European and international level, it will be possible to adapt policies, review processes and organizational setups in industry, such that AI technologies will be used responsibly and ethically in the future.

References

1. Russel S. and Norvig P.: Artificial Intelligence: A Modern Approach, 4th edn. Pearson Series in Artificial Intelligence, USA (2020)
2. Morgan F. & al: Military Applications of Artificial Intelligence: Ethical Concerns in an Uncertain World, https://www.rand.org/pubs/research_reports/RR3139-1.html, last accessed 2021/08/23
3. Parly F.: Intelligence artificielle et défense, Saclay (2019)
4. DGA: Guide de recommandations pour la spécification et la qualification de systèmes intégrant de l'intelligence artificielle, version 2 (2020)
5. Defense Innovation Board: AI Principles: Recommendations on the Ethical Use of Artificial Intelligence by the Department of Defense, https://media.defense.gov/2019/oct/31/2002204458/-1/-1/0/dib_ai_principles_primary_document.pdf, last accessed 2021/08/23
6. US Department of Defense: DOD Adopts Ethical Principles for Artificial Intelligence, <https://www.defense.gov/Newsroom/Releases/Release/Article/2091996/dod-adopts-ethical-principles-for-artificial-intelligence/>, last accessed 2021/08/23
7. Airbus: The Responsible Use of Artificial Intelligence in FCAS – An Initial Assessment, <https://www.fcas-forum.eu/en/articles/responsible-use-of-artificial-intelligence-in-fcas>, last accessed 2021/08/23
8. High-Level Expert Group on Artificial Intelligence (AI HLEG): Ethics Guidelines for Trustworthy AI, <https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai>, last accessed 2021/08/23
9. High-Level Expert Group on Artificial Intelligence (AI HLEG): The Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment, <https://futurium.ec.europa.eu/en/european-ai-alliance/document/ai-hleg-assessment-list-trustworthy-artificial-intelligence-altai>, last accessed 2021/08/23
10. Anderson S.: Target Classification, Recognition and Identification with HF Radar, In: RTO-MP-SET-080 Symposium on Target Identification and Recognition Using RF Systems (2004)
11. Technology readiness levels (TRL), Extract from Part 19 - Commission Decision C(2014)4995, https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf, last accessed 2021/08/23
12. ENISA's Ad-Hoc Working Group on Artificial Intelligence Cybersecurity: AI Threat Landscape, <https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges>, last accessed 2021/08/23
13. Wikipedia The Free Encyclopedia, Precision and Recall, https://en.wikipedia.org/wiki/Precision_and_recall, last accessed 2021/08/23
14. European Parliament and the Council of Europe: Proposal for a Regulation of The European Parliament and of the Council laying down harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206>, last accessed 2021/08/23

Aeronautical ad-hoc networking based on artificial intelligence minimizing interference on ground networks

Léonard Caquot, Tristan Charrier, Badre El Bezzaz Semlali,
Oudomsack Pierre Pasquero, and Alexis Bazin

Direction Générale de l'Armement - Maîtrise de l'Information (DGA-MI), French
Ministry of Defense (MoD), DGA-MI - BP7 - 35998 Rennes Cedex 9, France

Abstract. The massive deployment of wireless communication systems generates an increasingly important congestion of the frequency spectrum. Most of the communication systems are located on ground. However, developments and investigations are in progress to improve the performance and increase the density of communication systems in other environments such as aeronautics. In this paper, we propose networking methods for aeronautical communication systems based on the use of frequency bands initially allocated to networks on ground. The objectives is to optimize the transmission quality of the aeronautical network while not causing interference on the networks on ground. We present two approaches based on artificial intelligence. The first is an exact approach using answer set programming, modeling the problem with constraints and using a generic solver to find a solution. Since this approach does not scale well, the second is an approximate approach using message passing neural networks and reinforcement learning, particularly suited for learning on graph data.

1 Introduction

The development of wireless communication systems during the last decades has been the key enabler for the connected world in which we live today. The deployment of communication systems is not limited to terrestrial applications. The need for effective data links for aeronautical applications becomes significant in a context of a more and more congested frequency spectrum [1–3].

In this article, we focus on aircrafts ad-hoc networks using frequency bands initially allocated to terrestrial networks. We consider each aircraft has several directive antennas. For each antenna, the pointing direction, in which the signals are transmitted or received, is configurable by the communication system.

To optimize the connectivity, we need first to build a mesh network by pointing all or a subset of each aircraft's antennas toward the other aircrafts. Second we need to build the maximum number of RF links within the network to ensure reliability. Third, we need to minimize the network diameter, which is the maximum number of relays between two aircrafts.

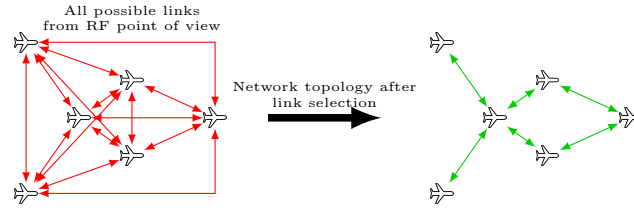


Fig. 1. Network topology construction

From these needs, we can already make some observations. The number of network possibilities increases quickly with the number of aircrafts. The complexity here is growing with $O(2^n)$, n being the number of inter-aircraft links in the network. Also, several routing paths are possible within each network solution to connect two aircrafts. Finally, seeking to optimize the network throughput or connectivity could be at the cost of a high interference on ground networks. The best transmitting antenna and the best routing path to optimize the communication performance (in terms of latency, data rate) do not guarantee the absence of interference on terrestrial networks.

To compute interference from the aeronautical network to terrestrial networks, several parameters have to be considered: the transmitting power, the directivity and radiation pattern of the antennas, the pointing direction of the transmitting antennas, the choice of the antenna to transmit the signal, the routing path followed by each data stream, using other aircrafts as relays or not. In what follows, we consider only a few parameters: the antenna transmitting and receiving the signals and the routing path followed by the data stream.

Thus, we cannot afford to evaluate all possible network topologies. Brute force algorithms quickly show their limits and can be used only in some very simple cases (with low number of aircrafts and/or a very low number of antennas per aircraft). Artificial Intelligence (AI) methods are good candidates for this kind of networking problem. We use Answer Set Programming (ASP) [4], in particular the solver Clingo [5], where we describe which solutions are correct instead of writing an algorithm to exhibit the solution. Such an approach was used for nurse scheduling [6], team-building at the seaport of Gioia Tauro [7], collaborative housekeeping of robots [8]...¹

Yet, such an approach does not scale well with the number of aircrafts. Thus, we also study the use of neural networks. It has many advantages when it comes to solving high combinatorial problems in an approximate way [9]. Some new neural networks architectures have been developed recently, called Message Passing Neural Network [10] (MPNN). This technique makes it possible to use neural networks on graph data such as antenna/aircraft graphs in our case. It solves two major issues: dealing with variable data size (number of nodes and/or edges in a

¹ See <https://link.springer.com/article/10.1007/s13218-018-0548-6> for an overview of industrial applications of ASP.

graph), and capturing local topology information within nodes neighborhoods. Yet, learning such models is usually done on a static set of input data. Over the last few years, deep reinforcement learning [11] (RL) has emerged and have shown great accomplishments in dynamic settings. A notorious example of RL is AlphaGo Zero [12]. It is the AI that beat the Go champion Lee Sedol in 2017.

The article is organized as follows. In Section 2, we define the computational problem. In Section 3, we detail the use of ASP to solve the problem. In Section 4 we detail ongoing work mixing message passing neural networks and reinforcement learning to have a scalable approach. Finally in Section 5 we give some conclusive remarks.

2 Stating the problem

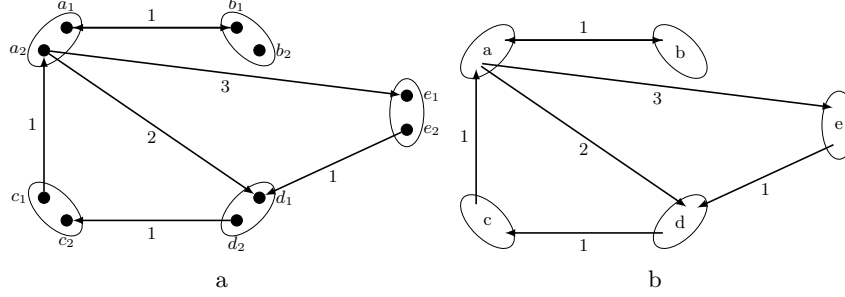


Fig. 2. Example of network with its antenna graph (subfigure a) and corresponding aircraft graph (subfigure b).

Definition Figure 2a. gives an example of network made of aircrafts labeled a to e . Each aircraft has two antennas labeled 1 and 2. For instance aircraft a has two antennas a_1 and a_2 . In our leading example, we assume that all the antennas are omnidirectional. Thus, each antenna can communicate with any other one. We also assume the received power signal depends on the distance. We symbolize the interference on ground networks with a value on the links. The greater the value, the higher the interference level the link generates. Thus, the link $a_1 \rightarrow b_1$, that has a value of 1, generates a lower interference on ground networks than $a_2 \rightarrow e_1$, that has a value of 3.

Each aircraft can communicate with any other one but not necessarily directly. In Figure 2, if aircraft c needs to communicate with b , it has to use at least one a as a relay. More precisely, two links are used: $c_1 \rightarrow a_2$ and $a_1 \rightarrow b_1$. The network is not optimal for interference on ground since we have the link $a_2 \rightarrow e_1$ generating heavy interference.

We see that the graph from Figure 2a. differs from usual graphs, since there are two types of nodes: aircrafts and antennas. We call such graphs *antenna*

graphs. Abstracted graphs where only links between aircrafts are drawn and antennas are discarded are called *aircraft graphs*. For instance in Figure 2b. we draw the aircraft graph corresponding to the antenna graph in Figure 2a. Formal definitions for both types of graphs are given in Appendix.

We say that there is a path between two aircrafts in an antenna graph if there is a path between these aircrafts in the corresponding aircraft graph. An antenna graph is said *strongly connected* if the corresponding aircraft graph is strongly connected in the classical sense of graph theory, i.e. there exists a path between any pair of aircrafts.

Now we need some metrics to evaluate the relevance of a given antenna graph. The three metrics we consider are the *cost*, i.e. the maximum value of all the interference values present in the edges of the graph; the *number of edges*; the *diameter*, i.e. the length of the longest acyclic path.

For instance, for the graph of Figure 2, the cost is 3 (since the edge with the highest cost is (a_2, e_1)), the number of edges is 7 (since (a_1, b_1) goes both ways) and the diameter is 4 (since the most distant aircrafts are e and a with a smallest path of length 4).

Formally, the optimization problem we solve is the following. In input, we consider a strongly connected antenna graph G . We output a strongly connected antenna graph G' that is a subgraph of G minimizing the interference, the number of edges and the diameter in any order. Usually, the graph in input is a complete graph where every edge is present. In practice, some edges can be absent if some links cannot be physically established between some couples of antennas.

Minimizing all metrics at once is impossible since reducing the diameter involves increasing the number of edges and vice versa. Since the main focus here is avoiding interference, the first metric we minimize is the cost. Then we minimize either the number of edges or the diameter.

3 Exact solving using answer set programming

We present in this section an algorithm for solving exactly this problem. The main idea is to use answer set programming (ASP) and more specifically the solver Clingo. This way, we only need to specify our problem as an input for Clingo and let the solver find the best solution.

Modeling the problem Yet, we cannot have float values for costs since ASP only works on discrete problems. It is not an issue here because we can directly know what is the minimal cost we can have by iterating over sorted edges until we find a strongly connected subgraph. When this cost is found, we remove all edges that have strictly higher cost and we give the corresponding subgraph as input to find the best number of edges/diameter.

The main ingredients when writing a description in ASP are *facts*, *rules*, *constraints* and *objectives*.

Facts They represent the initial data. First, the predicate `antenna(p, a)` means that the antenna a is on aircraft p . Both are given by numbers in the ASP paradigm. Then the predicate `edge(a1, a2)` means that the edge (a_1, a_2) between antennas a_1 and a_2 is available. Finally the subgraph choice:

`{chosen(A1, A2) : A1 = 0..9, A2 = 0..9, edge(A1, A2)}.`

The braces mean that we are free to choose whether $chosen(a_1, a_2)$ is true or false. Such a choice is available for any a_1 and a_2 between 0 and 9 such that $edge(a_1, a_2)$ is true (9 is the number of antennas minus 1). Basically, we choose some edges in the graph, which describes a subgraph.

Rules They describe inferences we can make about initial data. Here the rules are used to describe the existence of paths between two nodes.

(A) `path(P1, P2) :- antenna(P1, A1), antenna(P2, A2), chosen(A1, A2).`
 (B) `path(P1, P2) :- antenna(P1, A1), antenna(P3, A3), chosen(A1, A3), path(P3, P2).`

It means that there exists a path between p_1 and p_2 if there is a direct edge between them (rule A) or if we can find a direct edge between p_1 and p_3 such that p_3 has a path with p_2 (rule B). We similarly define the distance between two aircrafts `distance(N, P1, P2)`, true when p_1 and p_2 are at distance n .

Constraints They filter some possible solutions that are not correct. Here we do not want any subgraph but only strongly connected ones. With the predicate `path` previously defined it is fairly easy to define strong connectivity.
`:- not path(P1, P2), P1 = 0..4, P2 = 0..4, P1 != P2.`

It means that it is forbidden to find p_1, p_2 such that $p_1 \neq p_2$ and `path(P1, P2)` is false, i.e. a pair of aircrafts that do not have any path between them. Such a constraint is the exact definition of a strongly connected subgraph.

Furthermore when we want to focus on minimizing the diameter of the graph, we first compute the diameter D of the initial graph with a dedicated script. Then we know that the minimal value of the diameter of any subgraph is D , expressed by the following constraint: `:- distance(D+1, P1, P2)`. It means that it is forbidden to find two aircrafts such that the distance between them is $D + 1$.

Objectives Finally the objective is to minimize the number of edges.
`#minimize {1@1, A1, A2 : chosen(A1, A2)}.`

Results To evaluate the approach we take as input a complete antenna graph with random costs in the edges. Each aircraft has a random number of antennas between 4 and 8. We run 100 simulations per number of aircrafts and timeout a run in 300 seconds.

The first experiment corresponds to the minimization of the cost, then the number of edges, then the diameter. The results are given in Figure 3a. We represent the execution time with and without extremal values, and the number of timeouts. We see that up to 40 aircrafts the solving time is low in most cases. In cases when the solver takes more time, it is actually ensuring the optimality

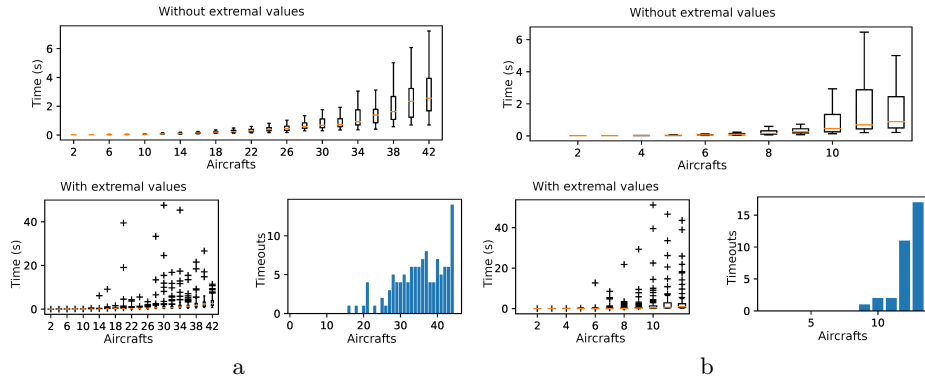


Fig. 3. Experimental results for minimizing in order cost, number of links and diameter (subfigure a); cost, diameter, number of links (subfigure b).

that takes time, not finding a solution. We can interrupt the solver in these cases to have a non-necessarily optimal solution.

The second experiment corresponds to the minimization of the cost, then the diameter, then the number of edges. The results are given in Figure 3b. It is a much more challenging task, the solver begins to be stuck for more than ten aircrafts.

The combinatorial explosion makes the problem hard to solve, thus this approach is satisfying for a low number of aircrafts/antennas. Yet this approach does not scale well. Therefore, we need other methods in real time contexts.

4 Approximate solving using reinforcement learning and Message Passing Neural Networks

We propose an approximate method based on reinforcement learning (RL) using message passing neural networks (MPNN). It should scale better to a high number of aircrafts/antennas.

Problem formulation RL models learn how to act in a dynamic environment by executing actions and gradually tuning their strategies with the feedback of the environment, the *reward* (for instance in games by seeing when they win or lose). Each scenario executed from start to end is called an *episode* and a given state in the episode is called a *step*. Since the RL model acts in the environment and adapts itself, it is called an *RL agent*.

As reinforcement learning and message passing neural networks are both quite challenging artificial intelligence techniques, our priority was first to demonstrate the feasibility of joining those two methods to solve the problem of graph generation under constraint. To do so, we have defined a toy problem that makes both development and interpretation easier. It is defined as follows.

- *Initial configuration:* A random antenna graph G_0 is generated at startup. A 2D position (x, y) is randomly assigned to each node of the graph, such that antennas are close to their aircrafts. The cost of an edge is defined as the euclidean distance between its two nodes, or zero if they are not connected. A new initial configuration is generated at each new episode, and then will be iteratively modified at each step of the episode by the RL agent.
- *Steps:* During an episode, the agent changes the graph by switching its edges from "on" to "off" or from "off" to "on". Selecting an edge takes two consecutive actions of the agent. One step to select the first node of the edge, and one to select the other. The process of selecting a node involves both MPNN module and RL module.
- *Reward:* The reward function is a weighted sum of the number of connected components and the cumulated edge distance. Those two opposite objectives make this toy problem relevant to our main problem defined in Section 2, and also leads to a challenging learning process.
- *Update:* As each graph can be evaluated thanks to a reward function, the RL agent can then be trained to generate more and more near-to-optimal graphs. More precisely, the agent is trained to map nodes embeddings to good nodes scores, that leads to better choices of edges when it comes to switching an edge state of the graph. Furthermore, as both MPNN and RL modules are trained in the same time, the gradient backpropagation used in RL training leads not only to better nodes scores, but also to better graph and nodes embeddings.

Note that the way we choose edges to switch (by selecting two nodes in two steps) is scalable. Each step requires n_{nodes} neural network inferences, where n_{nodes} is the number of nodes in the original graph G_0 . Indeed, selecting one edge requires approximately $2n_{nodes}$ inferences. Another option would have been to benefit from the ability of MPNN to produce edges embeddings.

However, each step would then require approximately n_{nodes}^2 inference to get all edges scores. Our solution quickly becomes more interesting from a scalability perspective.

Implementation and results In our experiment, we generate graphs with the following initial conditions. The number of aircraft nodes is picked randomly between 2 and 10. The number of antenna nodes is picked randomly for each aircraft between 2 and 4. The positions (x, y) of aircrafts are picked randomly using uniform distribution. The antennas are located randomly on a circle of fixed radius around their aircraft. The number of initial activated connections between antennas is picked randomly using uniform distribution. Random connections are then activated one by one until this number is reached.

We then trained a TD-Advantage Actor Critic [13] algorithm, a type of RL agent, to learn to activate or deactivate connections.

Figure 4 shows the evolution of the sum of rewards during training. The fact that it increases as the training progresses demonstrates that the agent learns to

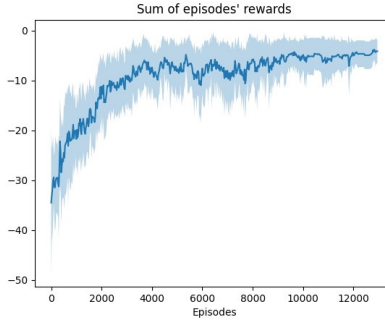


Fig. 4. Evolution of sum of episodes' rewards during training.

solve the problem properly. Manual check also confirmed that solutions produced by our model were relevant.

Our model takes around 6 hours to converge. This time could be reduced by using GPU instead of CPU during training, and by using more sophisticated RL algorithms such as PPO. It is also important to note that even if training process takes time, neural networks are really fast once trained (just a few milliseconds per step), making our solution very promising with regard to real-time issues.

5 Conclusion

In this work, we tackled the problem of networking several aircrafts while minimizing the interference on ground networks. We have studied two solutions based on artificial intelligence. We first focused on answer set programming, that provides and guarantees an optimal result. However, its computational cost makes it difficult to use above tens of aircrafts. Thus, it is hardly suited for both real time and large networks situations. Those limits led us to study another technique based on neural networks. The solution we developed mixes two innovative approaches: message passing neural networks and reinforcement learning. There is no guarantee of finding the optimal solution with this technique. However we have observed that such a technique can lead to good approximate solutions and does not suffer from the same combinatorial explosion as exact approaches.

As we used a toy-problem to more investigations are necessary to make it compatible with our real problem. Indeed, we simplified by assimilating the notion of interference to an inter-node distance. In order to train our model on more realistic data and bridge the gap with the real-life problem, we now need to add some physics simulations bricks to our reinforcement learning environment.

We also want to consider other constraints in a more distant future, such as temporal consistency in dynamic environments. To do so, stability metrics will have to be defined and added to our reward function. New models will then have to be trained using those new constraints, to get new good solutions. Those models could use recurrent neural networks to address those temporal issues.

References

1. Akram Hakiri, Pascal Berthou, Julien Henaut, Daniela Dragomirescu, and Thierry Gayraud. Performance evaluation of wireless sensor network for spatial and aeronautic systems. In *2010 17th International Conference on Telecommunications*, pages 879–886. IEEE, 2010.
2. Kimon Karras, Theodore Kyritsis, Massimiliano Amirfeiz, and Stefano Baiotti. Aeronautical mobile ad hoc networks. In *2008 14th European Wireless Conference*, pages 1–6. IEEE, 2008.
3. Hou Yan-lan et al. Diffserv in the aeronautic telecommunication network (atn) qos. In *2009 Integrated Communications, Navigation and Surveillance Conference*, pages 1–10. IEEE, 2009.
4. Vladimir Lifschitz. *Answer set programming*. Springer Berlin, 2019.
5. Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Sven Thiele. A user’s guide to gringo, clasp, clingo, and iclingo. 2008.
6. Carmine Dodaro and Marco Maratea. Nurse scheduling via answer set programming. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 301–307. Springer, 2017.
7. Francesco Ricca, Giovanni Grasso, Mario Alviano, Marco Manna, Vincenzino Lio, Salvatore Iiritano, and Nicola Leone. Team-building with answer set programming in the gioia-tauro seaport. *arXiv preprint arXiv:1101.4554*, 2011.
8. Esra Erdem, Erdi Aker, and Volkan Patoglu. Answer set programming for collaborative housekeeping robotics: representation, reasoning, and execution. *Intelligent Service Robotics*, 5(4):275–291, 2012.
9. Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.
10. Justin Gilmer, Patrick F Schoenholz, Samuel S Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.
11. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
12. David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.
13. Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000.

A Formal definition of antenna and aircraft graphs

We first define two types of graphs: *antenna* graphs (like Figure 2a) and associated *aircraft* graphs (like Figure 2b).

Definition 1. An antenna graph $G = (P, A, E, \omega, \delta)$ is a tuple where P is the set of aircrafts, A is the set of antennas, $E \subseteq A \times A$ is the set of edges, $\omega : E \rightarrow \mathbb{R}$ is the cost function, and $\delta : P \rightarrow 2^A$ is the function assigning each aircraft to its antennas. We impose the constraint that each antenna is in exactly one set $\delta(p)$.

The cost function corresponds to the contribution of the link to the interference on ground networks. In practice, it is provided by a matrix of scalar values, one value for each possible edge.

Example 1. The antenna graph of Figure 2 is $G = (P, A, E, \omega, \delta)$ with:

- $P = \{a, b, c, d, e\}$;
- $A = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, e_1, e_2\}$;
- $E = \{(a_1, b_1), (b_1, a_1), (a_2, e_1), (e_2, d_1), (d_2, c_2), (c_1, a_2), (a_2, d_1)\}$;
- $\omega(a_1, b_1) = \omega(b_1, a_1) = \omega(d_2, c_2) = \omega(c_1, a_2) = \omega(e_2, d_1) = 1$, $\omega(a_2, d_1) = 2$, $\omega(a_2, e_1) = 3$;
- $\delta(a) = \{a_1, a_2\}$, $\delta(b) = \{b_1, b_2\}$, $\delta(c) = \{c_1, c_2\}$, $\delta(d) = \{d_1, d_2\}$, $\delta(e) = \{e_1, e_2\}$.

Connectivity notions for antenna graphs are intrinsically linked with their corresponding aircraft graphs. Such graphs are defined as follows.

Definition 2. An aircraft graph $G_p = (P, E_p, \omega_p)$ is a tuple where P is the set of aircrafts, $E_p \subseteq P \times P$ is the set of edges and $\omega_p : E_p \rightarrow \mathbb{R}$ is the cost function.

The idea to go from antenna to aircraft graphs is to define the aircrafts as the main nodes and to keep edges between aircrafts that have linked antennas.

Definition 3 (From antenna to aircraft graphs). Let $G = (P, A, E_p, \omega, \delta)$ be an antenna graph. The corresponding aircraft graph is $G_p = (P, E_p, \omega_p)$ where:

- $(p_1, p_2) \in E_p$ if and only if there exists $\alpha_1, \alpha_2 \in A$ such that $\alpha_1 \in \delta(p_1)$, $\alpha_2 \in \delta(p_2)$ and $(\alpha_1, \alpha_2) \in E$;
- $\omega_p(p_1, p_2) = \max\{\omega(\alpha_1, \alpha_2) \mid (\alpha_1, \alpha_2) \in E, \alpha_1 \in \delta(p_1), \alpha_2 \in \delta(p_2)\}$.

MPC4SaferLearn: Privacy-Preserving Collaborative Learning with Secure and Robust Decentralized Aggregation [★]

Pierre-Elisée Flory and Katarzyna Kapusta

ThereSIS, Thales SIX GTS France, Palaiseau, France
surname.name@thalesgroup.com

Abstract. We present a novel framework for privacy-preserving collaborative learning that distributes the aggregation procedure between multiple participants. The confidentiality and the correctness of the aggregation are ensured by Multi-Party Computation techniques and preserved unless all of the participants are corrupted by an adversary. Therefore, the framework can be used for collaborative learning in use cases where participants do not trust each other. Our performance results show the practicability of the approach. We explain how to choose between various available Multi-Party Computation protocols and security models in order to balance between privacy, security, and performance requirements.

Keywords: Federated Learning · Multi-Party Computation · Privacy-Preserving Machine Learning

1 Introduction

Data is inevitably at heart of the Machine Learning (ML) life cycle as accuracy of ML models depends on the quality and the size of their training datasets. For many use cases, gathering a sufficient amount of valuable data for the training may be very difficult due to privacy or regulation reasons. For instance, a model for predictive maintenance of military equipment would strongly benefit from training on various data sources as Mean Time Between Failure varies depending on the environment of exploitation. However, centralizing data collected during military missions may be impossible as it could reveal sensitive information about the nature of each mission to the entity performing the training.

Federated (or Collaborative) Learning [11] address the lack of data by enabling the training of a model on various private datasets without the need of centralizing them. In more detail, local models are trained on local private datasets and only some information about these local models is centralized and

[★] *The research leading to these results has received funding from the European Union's Preparatory Action on Defence Research (PADR-FDDT-OPEN-03-2019. This paper reflects only the authors' views and the Commission is not liable for any use that may be made of the information contained therein.*

aggregated in order to construct the global model that will capture the knowledge of the multiple local models. In the classical version of FL, all local models share the same architecture and the global model is constructed using, for instance, a weighted average of the local model weights. Classical FL was shown to be very efficient in terms of the global model accuracy and transmission costs. However, its adoption in domains such as health care or military is slowed down by insufficient security and privacy guarantees. Indeed, even if the data are not explicitly exchanged, privacy attacks on the global model are still possible [4, 10]. Especially the central aggregator does not only know the architecture of the local models, but can also deduce information about their internals, such as weights or gradient. Moreover, the aggregator can be subject to intentional attacks or accidental failures.

To address the problem of securing collaborative learning on sensitive data, we introduce MPC4SaferLearn: a framework for collaborative learning that does not require a single central element and that provides higher privacy and confidentiality guarantees than the classical approach. We rely on an alternative approach to collaborative learning, in which only predictions of the local models are necessary during the aggregation [12], which reduces the risks of privacy leakage. Our main contribution is that in order to reinforce the learning against intentional attacks or accidental failures, we replace the central aggregator by a set of distributed aggregators performing computation on encrypted data. A user may then balance between performance, security, and privacy requirements.

2 Relevant works

2.1 Private Aggregation of Teacher Ensembles

Private Aggregation of Teacher Ensembles (PATE) [12, 13] is a privacy-preserving alternative to classical FL. Participants neither have to use the same model architecture nor to share information about their local models' parameters. Instead, participants of the learning train a global model in a semi-supervised way by labeling a limited number of public data using private classifiers. More precisely, each participant uses its local model to infer a label for each sample of a public dataset and shares the results of the labeling with a central aggregating server. This server combines labels coming from different sources: for each sample of the dataset, the aggregator selects the label that was chosen by the majority of private classifiers. It adds noise to the aggregation in order to provide privacy in a situation where a consensus is not attained between participants. In PATE, the adversary has access to the final model but not to the aggregator.

Secure Private and Efficient Deep Learning (SPEED) [5] secures PATE against an honest-but-curious aggregator that would like to infer information about the local models by observing inputs provided by the participants. Labels are encrypted using Homomorphic Encryption (HE) and the aggregation is performed in the encrypted domain. In order to address the problem of an aggregator that could not be trusted with the noise generation, the noise is generated by the participants. In SPEED, the adversarial model supposes that the adversary has

access not only to the final model but also to the aggregator: it can impact the noise generation procedure but cannot change the aggregation procedure.

2.2 Multi-Party Computation

Multi-Party Computation (MPC) [9] is a set of cryptographic techniques that enables a group of ‘parties’ to collaboratively perform a computation, even if they do not fully trust one another. More precisely, each party is assumed to hold some private data that they do not want the other parties to learn. With MPC, the parties can use their private data in a computation in such a manner that each party does not learn anything else than the computation result. An important benefit of MPC is that it does not require the existence of a trusted third party.

An MPC protocol is considered secure if it doesn’t leak more information than an ideal situation where a perfectly trusted third-party is involved. In particular, two elements are essential to determine the capabilities of a protocol: the number of members of the consortium that are corrupted by an adversary and the behaviour of the corrupted parties. Protocols are split in two between those making the assumption that the adversary cannot achieve to corrupt half the participants (*honest majority*) and those functioning even in unfavorable situation with a corrupted or *dishonest majority*.

What is implied by *corrupted party* depends on the assumed possible behaviour of the adversary. The most basic type of corruption is an *honest-but-curious* (also called *passive* or *semi-honest*) adversary. This means that the adversary follow the correct protocol but tries to extract information from the data it sees. Alternatively, the adversary can be considered *malicious* or *active* if no assumption on its behaviour is made. In particular, it can deviate from the protocol. Some refinement is provided by the *covert* behaviour that considers that the malicious adversary deviates from the protocol only if the probability of being caught is low.

The most common way of implementing MPC is to rely on secret-sharing. Secret sharing splits each input data into shares, a subset of which is required for data reconstruction. The computation is then performed on these data shares in a way that shares of an input are never gathered on a single machine.

3 Motivations

We replace the central aggregator in PATE with a distributed aggregator secured using MPC. Such an aggregator can solve three problems that a single aggregation server operating on encrypted data cannot address. The first one is that the secure aggregation server cannot be a participant of the computation: participant own the decryption key that the aggregation server should not possess. Likewise, an aggregation server is not supposed to collude with one or more participants. In MPC, a participant of the learning can also have the role of an aggregator. Second, while homomorphic-encryption prevents the server

from decrypting the data, it does not provide mechanisms to tackle a malicious server deviating from the protocol. For instance, we notice that an aggregation server corrupted by an active adversary can omit some of the inputs, modify the order of the computation results and/or alter the encrypted data as it is not authenticated. Homomorphic encryption could be enriched with additional expensive verification mechanisms [8] that would detect malicious behaviour of the aggregator. In contrast, MPC protocols addressing the active adversary already embed mechanisms that enable detecting deviations from the agreed protocol. Last but not least, MPC provides resilience mechanisms that enable to continue the training even if a majority of computing parties fail due to an attack, e.g., Denial-of-Service, or an unintentional problem.

4 MPC4SaferLearn: architecture overview

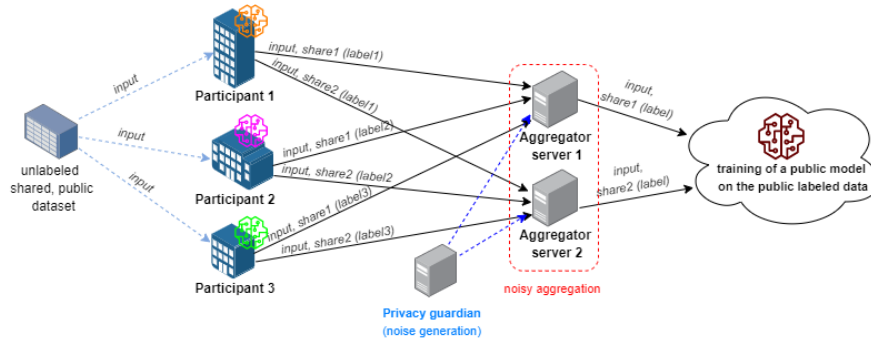


Fig. 1: *High-level overview of an example of the proposed architecture with three input providers and two aggregators.*

An example of the proposed architecture is presented in Figure 1, with three participants of the learning and two separate computing parties acting as the distributed aggregator. Note that the separation between the participants and computing parties could be only logical (we chose this setting for the sake of simplicity of the visual representation).

At a high level, we use the same noisy aggregation protocol as in PATE, only on encrypted data and in a distributed way. The exact exchanges between the participants and aggregators as well as between the aggregators themselves will depend on the choice of the MPC protocol.

The final result computed by the aggregators is also encrypted and distributed between them. The decryption of the result is done by gathering the shares of the results. This can be done by all the aggregators at the same time or by sending the shares of the result to selected participants or to a different end user that will train the final model. In Figure 1, we present the last option:

the results are reconstructed in the cloud to train the final model and make it available, for instance, in a Machine Learning as a Service mode. Such configuration would particularly fit a use case where the consortium would rather like to monetize their private data than improve their local models. Their participation in the labeling would then be paid and the predictions of the final model would be accessible to all MLaaS users.

For both performance and privacy reasons, we generate the noise outside the aggregators. More precisely, we propose two different ways of noise generation. Either each participant, when producing a prediction of a public sample, directly adds a calibrated noise to its private input (solution similar to the one presented in [5]) or we can use a so-called *privacy-guardian*: an external party not involved in the computations that work only as a source of random noise. While relying on a privacy guardian for the noise generation, the trust assumptions are particularly low since it has no access to the shares and little motivation for deviating from the protocol. Such involvement of external sources of randomness is a well known optimization trick used in multiple MPC implementations [2].

5 Performance Results

To implement the MPC-based aggregation performed between the aggregating servers, we used the MP-SPDZ [6] open-source framework that enables easy implementation and benchmark of multiple MPC protocols (multiple threat models and MPC techniques are covered in a total of 30 MPC variants). The computation is coded in a Python-like language, which is then compiled into VM bytecode depending on the MPC protocol used. This way we can quickly switch from one protocol to another, balancing between security and performance requirements for the same aggregation procedure. We used C++ to implement the client processing at the input providers side.

Protocol	Majority	Adversarial behaviour
Shamir [3]	Honest	Passive
Malicious Shamir [3]	Honest	Active
Semi [7]	Dishonest	Passive
Cowgear [7]	Dishonest	Covert
Mascot [7]	Dishonest	Malicious

Table 1: *Summary of the 5 MPC protocols used during the presented experiment.*

Table 1 shows an excerpt of the MPC protocols that have been used to run the secure aggregation. They were chosen as they each cover a different security model. The two Shamir-based protocols are historic protocols using Shamir’s Secret Sharing [3]. MASCOT [7] is a state-of-the-art protocol addressing the active attacker model, and Semi and Cowgear are variation on MASCOT that

improve performance by lowering security guarantees. Performance comparison between these protocols is shown in Figure 2.

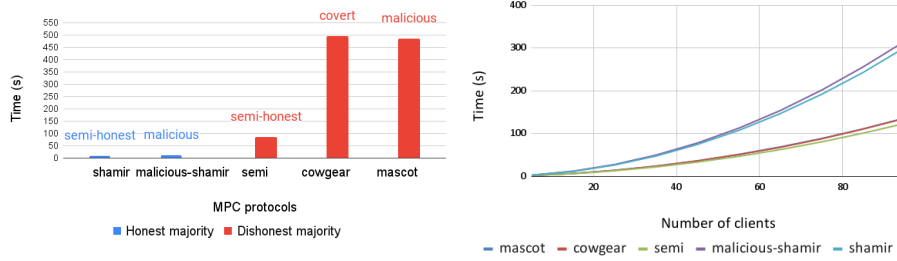


Fig. 2: *Performance comparison of different protocols. Left: Computation time comparison for aggregating 100 rounds of votes with 5 participants and 3 computing parties. Right: Computation time of 20 rounds of online phase for different protocols as a function of the number of clients providing inputs.*

Performance depends strongly on the security model. Protocols making assumptions on the adversary’s capabilities perform better than those securing computations against an active adversary (they require verification after each operation). Moreover, the number of computing parties (aggregating servers in our use case) has an impact on the performance. As the number of computing parties increases, so does the confidentiality of the input data as it is more dispersed. However, this adds high communication overhead (three times the communication for upgrading from two to three computing parties using the protocol MASCOT). We did most of our experiments using 3 computing parties to avoid the special 2-parties case where honest majority is not defined (there would not be any adversary). In the MPC domain, using 3 computing parties is actually the usual reasonable compromise between security and utility.

Modern MPC protocols, such as MASCOT, use two separate stages called *offline* and *online*. The offline phase is a precomputation done by the computing parties, during which correlated randomness in form of Beaver’s triples [1] is generated. These random triples are used during the online phase that comprises the actual computation. Such decoupling allows to speed up the actual processing as it is often assumed that the offline phase is done in anticipation. In Figure 3, we present performance results for the two phases in function of the number of parties. The online phase is linear in the number of computing parties. The generation of correlated randomness during the offline phase is highly affected by the number of computing parties. The throughput drops quickly with the number of servers. This mainly explain the reason for not involving all the participants in the MPC computations.

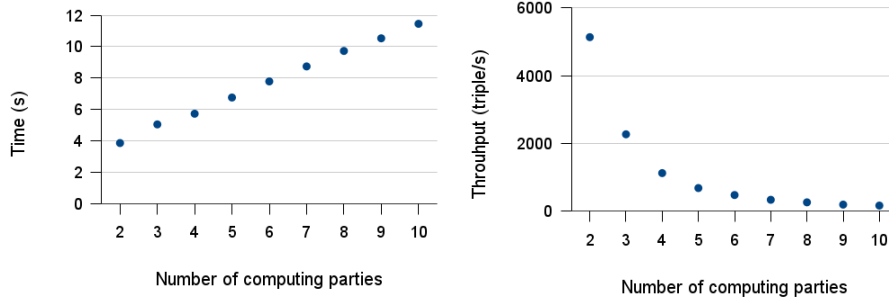


Fig. 3: *Effect of the number of computing parties on performances during the two computation phases. **Left:** Online phase duration (aggregation for 100 of inputs). **Right:** Correlated randomness (triples) generation throughput in offline phase.*

We observed significant performance improvements (see Figure 4) by aggregating labels in batch. Doing so reduces the number of communications, which is the main bottleneck in MPC, allowing overall faster aggregation. This should not be restrictive since we consider the public dataset as being indexed so that each participant can determine the order of the training data used.

Performances were measured on a Linux machine with a 4-th generation i5 processor running at 1.30 GHz using 16 GiB of RAM. The time for each aggregation is of the order of the second which correspond to a coherent order of magnitude for training a model. PATE-based collaborative learning doesn't require to run updates constantly but once in a while, which is acceptable given the computation time. The number of clients taking part in the aggregation as PATE teacher also affect the computing time, but as Figure 2 (right) shows, it is suitable for tens of clients, which meet current FL consortium requirements.

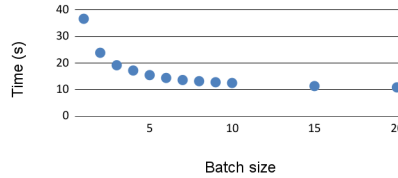


Fig. 4: *Comparison of computation time using different batch sizes (MASCOT protocol, 3 input providers, 3 aggregation servers, and 1000 inputs).*

The experiment carried out on multiple hosts gives a better understanding of the feasibility of the solution. In the safest security model, the secure aggregation requires 3 seconds for each training data. In a less constrained situation, with a

passive adversarial, the execution time drops at less than half a second per data. Future works include extended tests in real-world distributed environments and evaluating the usability of other kind of protocols such as garbled circuits.

6 Conclusions

We introduce MPC4SaferLearn, a framework for privacy-preserving collaborative learning that secures aggregation by distributing it over multiple participants or external providers. Thanks to the use of Multi-Party Computation, participants of the training consortium do not have to rely on any trust assumption regarding each others. They can easily balance between privacy, security, and performance requirements by adjusting the number of devices over which the computations are distributed or by switching between the multiple available MPC protocols. We believe that MPC4SaferLearn is thus well suited for sensitive applications such as collaborative learning on military or healthcare data.

References

1. D. Beaver. Efficient multiparty protocols using circuit randomization. In J. Feigenbaum, editor, *CRYPTO'91*, 1992.
2. C. A. Choquette-Choo, N. Dullerud, A. Dziedzic, Y. Zhang, S. Jha, N. Papernot, and X. Wang. Capc learning: Confidential and private collaborative learning, 2021.
3. R. Cramer, I. Damgård, and U. Maurer. General secure multi-party computation from any linear secret sharing scheme. Cryptology ePrint Archive, Report 2000/037, 2000. <https://ia.cr/2000/037>.
4. D. Enthoven and Z. Al-Ars. An overview of federated deep learning privacy attacks and defensive strategies, 2020.
5. A. Grivet Sébert, R. Pinot, M. Zuber, C. Gouy-Pailler, and R. Sirdey. Speed: secure, private, and efficient deep learning. *Machine Learning*, 2021.
6. M. Keller. MP-SPDZ: A versatile framework for multi-party computation. Cryptology ePrint Archive, Report 2020/521, 2020. <https://eprint.iacr.org/2020/521>.
7. M. Keller, E. Orsini, and P. Scholl. MASCOT: faster malicious arithmetic secure computation with oblivious transfer. In *ACM CCS*, 2016.
8. J. Lai, R. H. Deng, H. Pang, and J. Weng. Verifiable computation on outsourced encrypted data. In *Computer Security - ESORICS 2014*. Springer International Publishing, 2014.
9. Y. Lindell. Secure multiparty computation (mpc). Cryptology ePrint Archive, Report 2020/300, 2020. <https://ia.cr/2020/300>.
10. L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, and P. S. Yu. Privacy and robustness in federated learning: Attacks and defenses. *CoRR*, 2020.
11. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data, 2017.
12. N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017.
13. N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with PATE. In *ICLR*, 2018.

PRIVILEGE: PRiVacy and Homomorphic Encryption for Artificial IntElligence

Oana Stan¹, Vincent Thouvenot², Karel Hynek³, Romain Ferrari², Aymen Boudguiga¹, Renaud Sirdey¹, Alice Heliou², Tomas Cejka³, Katarzyna Kapusta², Daria La Rocca², Martin Zuber¹, George Vardoulas⁴, Ioannis Papaioannou⁴, Andreas Vekinis⁴ and Georgia Papadopoulou⁴

¹Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France `surname.name@cea.fr`

²Thales SIX GTS, 91120 Palaiseau, France, `surname.name@thalesgroup.com`

³CESNET z.s.p.o., Zikova 1903/4, 160 00 Prague, Czechia, `hynekkar@cesnet.cz`

⁴Intracom Defense, 21 km Markopoulou Ave., Koropi, GR 19441, Greece, `gepap@intracomdefense.com`

Abstract. PRIVILEGE solution advances the state of the art of defence technology, already using Artificial Intelligence (AI) systems, with respect to data security and privacy preservation in a collaborative setting. PRIVILEGE will strengthen collaboration among different allies on the secure analysis of sensitive defence and military data. The approach is based on combining distributed AI frameworks, such as federated learning and PATE, together with privacy-preservation and security tools such as Homomorphic Encryption, Verifiable Computation, or Multi-Party Computation. The proposed solution is universal, however, three specific real use cases will be targeted to validate the approach and demonstrate applicability of PRIVILEGE in practice: the classification of radio waves in defence operation, the classification of malicious network logs, and the video processing for unmanned vehicles.

Keywords: PATE, Federated Learning, Homomorphic Encryption.

1 Introduction

The progress in Artificial Intelligence (AI) and especially in Deep Learning has led to the widespread of these techniques in various application domains, including the defence and military fields. The AI-based tools and methods in the defence sector can be applied to various use cases, such as surveillance, autonomous weapons or vehicles, cybersecurity and anomaly detection. Since all these methods require access to very sensitive and critical data, usually in large volumes, several challenges still need to be addressed to easily and efficiently deploy AI algorithms in the defence context.

Moreover, even though it is an advantage for organizations to work collaboratively on the same AI system, data sharing between trusted partners in the context of defence projects is not a common practice today, principally because most of the data are usually classified. However, there are many situations in which data sharing between allied countries (e.g., in the context of the NATO alliance) would be beneficial, for example

for being able to share Intelligence Surveillance and Recognition data collected from unmanned vehicles allowing to recognize weapons, persons or mines in real-time.

The objective of the PRIVILEGE (PRIVacy and homomorphIc encryption for artificial intelliGence) project is to address the issue of secure collaboration of allied military forces without exchanging sensitive data. PRIVILEGE will design, deliver specifications and develop “privacy-by-design” machine learning training techniques.

More specifically, the goal of PRIVILEGE is to enrich AI with privacy-preserving techniques tools such as Homomorphic Encryption (HE) or Differential Privacy (DP), in order to be able to exploit confidential military and defence data throughout the life-cycle of AI methods with a focus on the learning step. Additionally, the novel privacy-preserving AI algorithms will be suitable for public domain (civilian) AI applications, making them GDPR compliant.

2 Context and motivation

Most AI techniques and models rely on unlimited access to large datasets and raise serious privacy concerns. Furthermore, these AI systems may be vulnerable to a growing body of statistical attacks, such as de-anonymization, model inversion, data extraction from model memory. Nevertheless, with growing public awareness and new privacy-focused laws, there is now an urgent need to tackle the privacy challenges in order to unveil the next generation of AI systems. This is even more urgent for regions of the world, such as Europe, that wish to combine strict citizen privacy protection policies and laws with the competitive advantage of developing novel AI technology.

In the defence and military domain, AI techniques are now applied in novel tools for surveillance, weapons control, autonomous driving and cybersecurity. All these applications require a large volume of training data, and the sharing of data between allies (e.g. in the context of NATO) may be useful even if it is difficult, and in some cases impossible. The data required by military applications are often classified since it may be exploited for behavior analysis and subsequent defence tactics disclosure.

In this context, the PRIVILEGE project’s aspiration is to allow allies to collaborate easier for the exploitation of various defence tools that rely on AI methods without disclosing their confidential data. For this matter, cryptographic techniques enabling secure computation - processing on encrypted data without its decryption - seem mature enough to be considered in the AI context. Fully Homomorphic Encryption (FHE), part of the provably-secure cryptography, allows encryption and general computation directly over encrypted data and has now a well-founded corpus of security properties. An interesting alternative to FHE is Multi-Party Computation (MPC). MPC distributes the encrypted computation across multiple servers, protecting the inputs privacy unless all of the involved parties collude. For long, FHE and MPC were considered too slow to be practical. In recent years, the application of these two techniques to real-world use cases has become more realistic thanks to the release of implementations optimizing their performance.

For now, most of the currently published work applies the FHE or MPC techniques to AI systems evaluating an already-learned public model over an encrypted private input. Moreover, these early endeavors were realized on very simple Artificial Neural Networks (ANN) as a first step to show the feasibility of this approach. However, the secure evaluation or training of real-world deep neural networks remains a challenge. The issues related to ensuring the privacy of network models or learning data is still an open problem. On the other side, for the ANN training, progress has been made with the design and democratization of various methods on training data from various sources such as the transfer learning, the Federated Learning or privacy-preserving ML frameworks.

The main technical target of PRIVILEGE is to address the data-privacy issues dealing with the collaborative training of Artificial Neural Network systems by means of secure computation and privacy-preserving techniques. This will allow multiple owners of learning datasets to build better models over the union of their training data without disclosing these datasets to one another nor to some third party.

3 Background

3.1 Federated Learning (FL)

The standard setting of machine learning considers a centralized dataset. However, centralizing data is not always possible as sending data may be too costly and data may be sensitive. Several parties may be interested in collaborating to train a machine learning model because the local dataset may be too small or biased. Federated Learning (FL) [1, 2] aims to train a machine learning model collaboratively while keeping the data decentralized. In distributed learning, used to train machine learning models faster, data are initially stored centrally and data distribution to individual computational nodes takes place afterwards. On the contrary, in FL settings, data are generated locally and thus initially distributed, they can be not independent nor identically distributed. A global model is computed in FL, averaging the local model trained on-device using the dataset owned locally by each worker (data-owner). There are three flavors of FL: Horizontal FL, Vertical FL, and Transfer FL. In the Horizontal FL, datasets composed of different samples share the same features, and the challenging point consists of aggregating the information from the data-owners for the training. We usually assume honest participants and security against an honest-but-curious server (it complies with the protocol honestly but can try to infer some information from the data it has). At the end of the training, the global model is exposed to all participants. Vertical FL occurs when the data-owners have data from the same or similar samples but with different features. Here, the main challenge is to aggregate the different features of the common samples across the data-owners in a privacy-preserving manner. At the end of the training, each data-owner holds only model parameters associated with its own features. Therefore, at inference time, the data-owners also need to collaborate to generate the output. Transfer FL applies to the scenario in which two datasets differ not only in samples but also in feature space. Only a small portion of the sample space and the feature space from

both parties overlap. In Privilege, we focus on Horizontal FL without assuming that the participants are honest.

3.2 Differential Privacy (DP)

Differential Privacy [3, 4] proposes a mathematical definition of Privacy. It states that for two adjacent datasets x and y , a random algorithm A achieves the (ϵ, τ) Differential Privacy if $P(A(x) = o) \leq \exp(\epsilon) P(A(y) = o) + \tau$, for all $o \in \text{Range}(A)$. Differential Privacy mathematically guarantees that anyone seeing the result of a differentially private analysis will essentially make the same inference about any individual’s private information, whether or not that individual’s private information is included in the input to the analysis. It provides a mathematically provable guarantee of privacy protection against a wide range of privacy attacks. It guarantees to protect only private information, not general information.

3.3 Private Aggregation Teachers Ensemble (PATE)

Private Aggregation of Teachers Ensemble [5, 6] proposes an ensemble approach and works by training local models on independent and identically distributed local datasets. First, a sensitive dataset is divided into several sub-datasets where some teacher models are locally trained. These teacher models are used to label a new public dataset, used to train a shared student model. To label the new public dataset, we use a noisy vote of the teacher models. The noise allows achieving Differential Privacy.

3.4 Homomorphic Encryption (HE)

Homomorphic Encryption schemes allow performing computations directly over encrypted data. With a Fully Homomorphic Encryption (FHE) scheme E , we can compute $E(m_1 + m_2)$ and $E(m_1 \times m_2)$ from encrypted messages $E(m_1)$ and $E(m_2)$. The first constructions of HE schemes, allowing either multiplication or addition over encrypted data, date back to the seventies. In 2009, against all expectations, Gentry [7] proposed the first Fully Homomorphic Encryption (FHE) scheme able to evaluate an arbitrary number of additions and multiplications over encrypted data, using the bootstrapping technique. Starting from Gentry breakthrough, many HE and FHE schemes have been proposed in the literature and are now classified into four “generations” based on their order of apparition, the underlying assumptions and the capabilities they offer. As such, the second generation schemes (e.g., BFV[8]) are allowing batching (packing several plaintext values into a single ciphertext and execution in a SIMD manner) and can be operated efficiently for certain applications in levelled mode, without bootstrapping. The third generation (e.g., TFHE [9]) are interesting for their fast bootstrapping operation, while the fourth generation (e.g., CKKS [10]) allow efficient rounding operations in an encrypted state.

As for the application of homomorphic encryption for AI systems and especially ANN, there are only a few recent works (e.g., Cryptonets [11]), and almost all concentrate on the private inference step using small public models on encrypted data.

3.5 Verifiable Computation (VC)

Verifiable computing enables a computer to offload the computation of some function to other untrusted clients while maintaining verifiable results. The general properties of Verifiable Computing have been extensively studied in the literature, and the field is particularly rich, including the use of Trusted Platform Modules, interactive proofs, Probabilistically Checkable Proofs (PCPs), efficient arguments, etc. In our work, we concentrate on the VC protocols that allow to check the integrity of the result of computations over homomorphically encrypted data. On this matter, Fiore et al. [12] propose a highly efficient VC scheme over homomorphically encrypted data for delegation of various classes of functions such as linear combinations or high-degree univariate polynomials or multivariate quadratic polynomials. More recent work on new VC protocols (e.g. [13]) goes beyond the degree-2 computation over homomorphically encrypted data. However, it requires a special hypothesis (such as a very large prime ciphertext modulus) and still lacks implementations.

3.6 Multi-Party Computation (MPC)

Multi-Party Computation [14] is a family of cryptographic techniques that enables a group of entities or ‘parties’ (i.e. people, organizations, devices etc.) to collaboratively perform a computation on some data, even if they do not fully trust one another. More precisely, each party is assumed to hold some private data that they do not want the other parties to learn. MPC enables the parties to perform a computation with this private data as input such that each party does not learn the other parties’ private input data. MPC can be an interesting alternative to FHE for some of the use cases. It allows to distribute computations between participants of collaborative learning and therefore remove the need of a central element, such as central aggregation server in FL. It can also provide protection against sophisticated adversaries that are able to corrupt multiple participants of the learning and make them deviate from the agreed protocol. However, the benefits of MPC come at the cost of increased communication.

4 Privacy and Homomorphic Encryption for Artificial Intelligence for the defence: PRIVILEGE solution

4.1 Science-to-technology innovative solutions

As stated in the Section 3.4, the recent work about applying homomorphic encryption to AI methods concerns mainly the prediction of Convolutional NN on homomorphically encrypted data. Public studies about using FHE in the training phase for neural nets are virtually nonexistent. As such, the design of a private-by-design learning step for collaborative algorithms such as Federated Learning and PATE requires the adaptation of the existing homomorphic encryption schemes, as well as the design of specifically tailored homomorphic learning methods.

Another breakthrough of the project is the design of an execution integrity layer for the aggregation function through Verifiable Computing. The use of Verifiable Computing for attesting the integrity of an aggregated learning result is an innovative idea that was not previously investigated in the open literature (see Section 3.5). This will be the first attempt to verify the integrity of the aggregated homomorphic encrypted result of Federated Learning through Verifiable Computing protocols.

Another significant advancement of PRIVILEGE is the use of Differential Privacy methods for collaborative learning in an application context. Currently, the deployment of such approaches is still nascent. The implementation, application and validation of DP mechanisms for the three PRIVILEGE use-cases (see section 5) will be a significant step forward. Moreover, its combination with homomorphic encryption in the context of Federated Learning and PATE will be an important achievement towards secure AI tools, concerning both training data and model confidentiality.

Finally, in contrast with the current state of the art solutions, which are mainly tested with the textbook example of the MNIST dataset (handwritten digit database) or some variants of it, PRIVILEGE will demonstrate the practical performances of the privacy-preserving framework for collaborative learning on three real use-cases.

4.2 Global view of the proposed solution

PRIVILEGE framework allows multiple dataset owners to build neural network models over the union of their training data without disclosing these data to others. This collaborative learning framework will be instantiated via Federated Learning and PATE approaches enhanced with various security methods – FHE, DP, VC and MPC providing countermeasures for various threats.

For the Federated Learning, DP techniques will ensure data privacy against threats coming from other parties participating in the learning step. The AI systems (even in a collaborative context) can leak information about the training instances. An attacker can use the outputs of the models or other information used during the training (such as gradients) to infer information about the datasets. As a result of using DP tools, attacks such as attribute inference attacks, property inference attacks or membership inference attacks can be avoided. Additionally, the HE technique can be a meaningful countermeasure against an honest-but-curious central aggregation server, which might reveal information about the global model. The VC can be used to provide integrity guarantees in the case of a malicious central aggregation server, which may not comply with the protocol and not correctly execute the federated averaging algorithm. Finally, MPC may be used to remove the need for a central aggregation server and thus distribute the trust across the participants of the Federated Learning.

Fig. 1 depicts a round of the learning process for FL with Federated Averaging where homomorphic encryption is used (in a simple single key setting) in the case of an honest-but-curious server. We assume that, before the training begins, the central server shares the topology of the used neural network model with all the clients, and the key generation and the key distribution takes place. Each of the M clients selected randomly to participate in a round has the same pair of secret and public homomorphic keys (sk , pk). The central server holds only the pk required for the homomorphic evaluation.

During one round, each client runs several epochs of minibatch stochastic gradient descent minimizing a local loss function. Once the clients perform this local training, they encrypt their local models and send them to the central server. The latter will perform a weighted averaging of the updated local models to obtain an encrypted updated global model. The current iteration ends by sending the global model to each of the M clients that decrypts it with the sk .

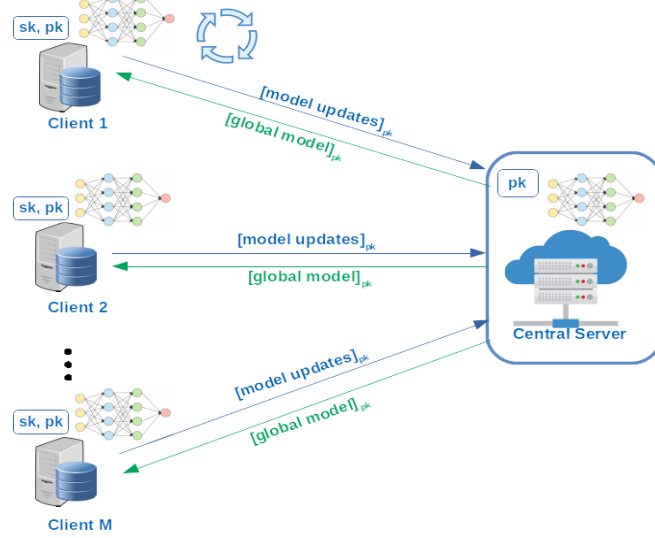


Fig. 1. A round of Privacy-Preserving Federated Learning with Homomorphic Encryption

As for the PATE framework, the DP technique will ensure a valid countermeasure against threats coming from end-users, while HE can be used in the case of an “honest-but-curious” server doing the labelling based on the noisy votes of the teachers.

5 Use Cases

An example of the application scenario that would benefit from deploying a PRIVILEGE solution comes primarily from cybersecurity. Two or more actors have separate databases of cybersecurity signatures that occurred on their customers' networks. It seems evident that building a model using a more extensive set of signatures can lead to improved detection capabilities. However, since these databases can carry companies' trade-secrets and contain private data, they cannot be disclosed (notwithstanding legal barriers by doing so).

For the validation and testing for real-life applications, the project will target three defence use cases brought by the partners: the classification of radio waves for antennas, the classification of malicious network logs¹ and the dissemination of intelligence surveillance and recognition data collected from unmanned vehicles.

¹ <https://nerd.cesnet.cz>

6 Conclusion

This paper presented a unique concept of an enhanced collaborative learning technology that includes privacy-preserving techniques. The principle was described as the PRIVILEGE solution, which combines advanced cryptographic tools (Differential Privacy, Homomorphic Encryption, Verifiable Computation and Multi-Party Computation) with collaborative learning frameworks such as Federated Learning and PATE. This approach will be initially validated using three real-world defence use cases, even if we believe its applicability is much wider. PRIVILEGE will allow end-users to benefit from combining independent, sensitive, not-disclosable local datasets and to build an AI model that will be shared among collaborating parties by keeping the privacy of data as a main priority.

Acknowledgments

The research leading to these results has received funding from the European Union's Preparatory Action on Defence Research (PADR-FDDT-OPEN-03-2019). This paper reflects only the authors' views and the Commission is not liable for any use that may be made of the information contained therein.

References

1. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., & He, B., A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection, 2019.
2. Kairouz et al., Advances and Open Problems in Federated Learning, 2019
3. Dwork, Differential Privacy, 2006
4. Dwork, & Roth. The Algorithmic Foundations of Differential Privacy, 2014
5. Papernot, Abadi, Erlingsson, Goodfellow, Talwar, Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data, 2016
6. Papernot, N, et al. Scalable Private Learning with PATE, 2017
7. C. Gentry, "Fully homomorphic encryption using ideal lattices", STOC, 169-178, 2009.
8. J. Fan et al., "Somewhat practical fully homomorphic encryption". IACR ePrint 144, 2012.
9. I. Chillotti, N. Gama, M. Georgieva, M. Izabachène, "Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds", ASIACRYPT, 3-33, 2016.
10. J. H. Cheon, A. Kim, M. Kim, and Y. S. Song, "Homomorphic encryption for arithmetic of approximate numbers", ASIACRYPT, 409-437, 2017.
11. N. Dowlin, et al., "CryptoNets: applying neural networks to encrypted data with high throughput and accuracy", ICML, 201-210, 2016.
12. D. Fiore, et al. "Efficiently verifiable computation on encrypted data" ACM SIGSAC Conference on Computer and Communications Security, 844-855, 2014.
13. D. Fiore, A. Nitulescu, D. Pointcheval. "Boosting Verifiable Computation on Encrypted Data". PKC2020
14. Y. Lindell. Secure Multiparty Computation (MpC). <https://eprint.iacr.org/2020/300.pdf>

Monte Carlo Tree Search for Multi-function Radar Task Scheduling

Marc Vincent^{1,2}, Amal El Fallah Seghrouchni^{2,3}, Vincent Corruble², Narayan Bernardin¹, Rami Kassab¹, and Frédéric Barbaresco¹

¹ Thales Land and Air Systems, 91338 Limours, France
`marc.vincent@thalesgroup.com`

² LIP6, Sorbonne Université, 75252 Paris, France

³ AI Movement – International Artificial Intelligence Center of Morocco, University Mohammed VI Polytechnic, Rabat, Morocco

Abstract. Multi-function radars require efficient resource management strategies to fulfill their missions. In particular, task scheduling is crucial to mitigate difficult situations such as when all tasks cannot be accomplished. However, current approaches may prove insufficient in the face of emerging threats. In this article, we present a new formulation for the scheduling problem. Our model allows building schedules in a flexible way, which facilitates the discovery of high-value solutions using heuristics or tree search. We show that our algorithms provide noticeable performance improvement over similar methods proposed previously.

Keywords: Artificial Intelligence · Multi-function Radar · Combinatorial Optimization · Scheduling.

1 Introduction

Multi-function radars (MFR) are a class of radars that are able to concurrently perform a range of functions that would otherwise have to be carried out by several distinct radars. MFRs have been made possible by the development of phased-array antennas, which enhance the radar’s flexibility by enabling greater control over waveforming and beamforming. An MFR’s functions usually include search, tracking, and various combat-assistance roles such as missile guidance. Given that a radar runs under constraints of time, power, and processing, a key challenge of MFR design is radar resource management, which consists in allocating these resources between functions [7]. Resources must be allocated according to the radar’s mission, which determines the priority level of each function.

Among these resources, time budget is the most critical, as the time spent performing a given task closely reflects its importance. Each function consists of one or more tasks, for example tracking a specific target; each task is carried out by performing a number of dwells characterized by requirements in desired execution time, duration, and power. Based on the current situation, the radar continuously generates dwell requests corresponding to its different functions.

The resulting list of requests is regularly transmitted to a scheduler, whose role is to decide which dwells should be executed and at what time. Efficient scheduling is crucial in overload situations, which happen when temporal constraints prevent from executing all dwells, forcing the scheduler to drop some of them according to their level of priority.

This scheduling problem is usually tackled with heuristics, yet such methods become increasingly suboptimal as the complexity of the situations radars face grows. These situations involve new threats, like drone swarms, which are likely to create overload, and hyper-maneuvering, high-velocity, and furtive targets, which might require more resources to track efficiently. Proposed alternatives make use of a range of methods, including expert systems, metaheuristics, and neural networks [5, 7, 10, 11].

In this work, we treat task scheduling as a sequential decision problem. In recent years, in the wake of significant progress in the field of combinatorial games, such as for Go [9], this kind of approach has been extended successfully to “single-player” combinatorial settings [6]. We build upon the work of Gaafar et al. [4], who first proposed to apply this type of technique to radar task scheduling.

First, we present our main contribution, a formulation of the radar task scheduling problem as a Markov decision process that allows building schedules in a flexible way. Then, we exemplify the usefulness of our model with a heuristic and a variant of Monte Carlo Tree Search (MCTS) adapted to this formulation. Finally, we evaluate our algorithms and demonstrate the improvement in performance brought by our approach.

2 Framework

We formalize the problem of radar task scheduling similarly to Gaafar et al. [4]. As we mentioned above, our model of an MFR scheduler processes one fixed set of tasks¹ at a time. An updated set of tasks is received at regular intervals. In the meantime, no tasks may be added to the set that is being processed. Each task in a set $I = \{1, \dots, n\}$ is characterized by its temporal constraints and its priority level. The temporal constraints of a task i are defined by its length L^i , its start time T_s^i and drop time T_{dr}^i (respectively the earliest and latest date at which it can start executing), and its due time T_{du}^i , which is the desired execution time. In this work, for any given instance, these constraints are all held constant over the course of the scheduling process: we do not consider preemptive tasks (e.g. for dwell interleaving) or variable duration time. Our goal is to determine for each task i in I whether to schedule it ($x^i = 1$, else 0), and if so at what execution time t^i , or to drop it ($y^i = 1$, else 0). Scheduled tasks must be entirely contained in a temporal frame $[0, T_{max}]$. In order to arbitrate between tasks in case of conflicts, each task is ascribed a drop cost C_{dr}^i and a delay cost C_{de}^i which reflect its priority level. The drop cost is incurred only when the corresponding task is

¹ In the rest of this article, we will use the standard terminology for scheduling, where “task” refers to the basic elements of a schedule—in our case, radar dwells. It should not be confused with the radar tasks mentioned earlier.

dropped while the delay cost determines how much the difference between the actual and ideal execution times is penalized; this difference is an absolute value, unlike in [4] where due times were not distinct from start times. How to assign relevant values to these costs in an operational context is left to future work. We write instances of the problem as $P = \{(L^i, T_s^i, T_{dr}^i, T_{du}^i, C_{dr}^i, C_{de}^i) \forall i \in I; T_{max}\}$, and its (partial) solutions, or schedules, as $s = \{(x^i, y^i, t^i) \forall i \in I\}$. The objective is to minimize the sum of costs (or total cost) $C_P(s)$. We summarize the problem below, where \oplus represents an exclusive or:

$$\begin{aligned} \min C_P(s) &= \sum_{i \in I} x^i |t^i - T_{du}^i| C_{de}^i + y^i C_{dr}^i \\ \text{s.t.} \quad &\begin{cases} T_s^i \leq t^i \leq T_{dr}^i \quad \forall i \in I \\ t^i + L^i \leq T_{max} \quad \forall i \in I \\ t^i + L^i \leq t^j \oplus t^j + L^j \leq t^i \text{ if } x^i = x^j = 1, \forall (i < j) \in I^2 \\ t^i \in \mathbb{R}^+ \quad \forall i \in I \\ x^i, y^i \in \{0, 1\} \text{ with } x^i = 1 - y^i, \forall i \in I \end{cases} \end{aligned} \quad (1)$$

This problem can be solved to optimality with mixed-integer programming (MIP). Unfortunately, since problem (1) is in NP¹, the computation time for MIP grows exponentially with the number of tasks, making it prohibitive for radar applications. In order to give further insight into the structure of the problem, we offer its detailed formulation for MIP:

$$\begin{aligned} \min C_P(s) &= \sum_{i \in I} l_{de}^i C_{de}^i + (1 - x^i) C_{dr}^i \\ \text{s.t.} \quad &\begin{cases} \left\{ \begin{array}{l} T_s^i \leq t^i \leq T_{dr}^i \\ t^i + L^i \leq T_{max} \\ l_{de}^i \geq t^i - T_{du}^i \\ l_{de}^i \geq T_{du}^i - t^i \end{array} \right\} \quad \forall i \in I \\ \left\{ \begin{array}{l} t^i + L^i \leq t^j + M(n^{ij} + o^{ij}) \\ t^j + L^j \leq t^i + M(n^{ij} + 1 - o^{ij}) \end{array} \right\} \quad \forall (i < j) \in I^2 \\ n^{ij} = 2 - x^i - x^j \quad \forall (i < j) \in I^2 \\ x^i \in \{0, 1\}, t^i, l_{de}^i \in \mathbb{R}^+ \quad \forall i \in I \\ o^{ij} \in \{0, 1\}, n^{ij} \in \mathbb{R}^+ \quad \forall (i < j) \in I^2 \end{cases} \end{aligned} \quad (2)$$

where M is an arbitrarily large number such that $M \gg T_{max}$. Of interest in formulation (2) is the presence of binary variables o^{ij} in addition to x^i . The values of o^{ij} determine the order in which the scheduled tasks are placed. This highlights the fact that problem (1) can be subdivided in three successive sub-problems:

1. *inclusion*: determine which tasks to schedule;
2. *ordering*: determine the order in which these tasks should be scheduled;
3. *time setting*: determine the execution times of these ordered scheduled tasks.

¹ This can be proven via a polynomial reduction with the knapsack problem by setting $T_s^i = C_{de}^i = 0$ and $T_{dr}^i = T_{max}$ for all i in I .

Note that problem (1) can also be solved with heuristics like earliest start time first (EST) or earliest deadline first (EDF). These heuristics (and variations thereof) are common in radar resource management because they are fast and easy to implement. However they will often produce poor solutions, for two reasons: first, because they operate on the premise that the schedule must be built by adding tasks in chronological order; second, because they do not account for task priority.

By contrast, we aim to develop approximate algorithms for radar task scheduling that can approach optimal solutions while keeping reasonable computation times. Our first step is to model problem (1) as a Markov decision process (MDP). MDPs are the most common formalization of sequential decision problems. An MDP is defined by: a finite state space \mathcal{S} , a finite action space \mathcal{A} , a distribution over initial states $\mu : \mathcal{S} \rightarrow [0, 1]$, a transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, which gives the conditional probability $\mathcal{T}(s' | s, a)$ of transition to the next state s' given the previous state s and selected action a , and a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ that gives the reward associated with a transition. We can associate a strategy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ to an MDP, i.e. a probability distribution over which action to take in a given state. The objective is usually to maximize the value function, which is the expectation of the sum of rewards: $V^\pi(s_0) = \mathbb{E}_{a_t \sim \pi, s_t \sim \mathcal{T}} \left[\sum_{t=0}^T \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) \right]$ where $\gamma \in [0, 1]$ is the discount factor and T the final step of an episode. Once we have defined a suitable MDP, we will need an algorithm that outputs a strategy for this MDP, which we call the decision algorithm.

Following Gaafar et al. [4], we define a state as a partial schedule, where a number of tasks have been scheduled at certain execution times and a number of others dropped while respecting the constraints of (1). The only initial state s_0 for a given instance of the problem is the associated empty schedule, where no tasks are scheduled or dropped. We also define an action as the choice of one task to schedule in the set of available tasks $A = \{i \in I \mid x^i = y^i = 0\}$. Terminal states are states where no actions are available, that is, complete schedules, where all tasks are scheduled or dropped. We can define the reward as the cost difference between the two states involved in a transition, i.e. $\mathcal{R}(s, a, s') = C_P(s) - C_P(s')$. With $\gamma = 1$, the value function becomes $V^\pi(s_0) = \mathbb{E}_\pi [C_P(s_0) - C_P(s_T)]$, which corresponds to our initial objective of finding a schedule s that minimizes $C_P(s)$. In practice, in our algorithms, we reason directly on costs.

The choice of the transition function is the most crucial aspect of this formalization. Transitions must deterministically decide what execution times to set and which tasks to drop when a new task is added to a partial schedule.

The simplest option, used in [4], is to add tasks chronologically: any newly added task i is assigned an execution time greater than that of all previously scheduled tasks, and tasks that cannot be placed after i are dropped. However, this restricts the range of viable strategies, since the decision algorithm has to solve both the inclusion and the ordering sub-problems at the same time, while the transition function only deals with the time setting sub-problem.

Instead, we propose a transition model that allows building schedules in a more flexible way. Our idea is to model the transition function such that it solves both the ordering and the time setting sub-problems at each step. That is, given a set S of scheduled tasks, the transition determines the order of tasks and their execution times by minimizing the total delay cost:

$$\begin{aligned} & \min \sum_{i \in S} l_{de}^i C_{de}^i \\ \text{s.t. } & \left\{ \begin{array}{l} T_s^i \leq t^i \leq T_{dr}^i \\ t^i + L^i \leq T_{max} \\ l_{de}^i \geq t^i - T_{du}^i \\ l_{de}^i \geq T_{du}^i - t^i \end{array} \right\} \forall i \in S \\ & \left\{ \begin{array}{l} t^i + L^i \leq t^j + M o^{ij} \\ t^j + L^j \leq t^i + M(1 - o^{ij}) \\ t^i, l_{de}^i \in \mathbb{R}^+ \forall i \in S \\ o^{ij} \in \{0, 1\} \forall (i < j) \in S^2 \end{array} \right\} \forall (i < j) \in S^2 \end{aligned} \quad (3)$$

Given a partial schedule $s = \{(x^i, y^i, t^i) \mid \forall i \in I\}$, in order to schedule a new task $j \in I$, we solve sub-problem (3) for $S = \{i \in I \mid x^i = 1\} \cup \{j\}$. If there is no solution to the sub-problem, j has to be dropped. After a task is scheduled, all remaining tasks in A can be tested for dropping using this procedure.

Sub-problem (3) can also be treated with a MIP solver; however this requires finding the order and execution times of all tasks in S at every transition. This is inefficient, because with this transition model, when we move from a partial schedule s to a new one s' by adding a task i , the execution times in s are already optimal with regard to the tasks scheduled in s . This means that if for example i can be placed at its desired execution time without interfering with already scheduled tasks, then there is no need to recompute the execution times of these tasks. To exploit this and a number of other optimizations, we implement a custom solver for sub-problem (3) which we do not detail here due to space limitations: the key idea is to recursively generate and evaluate permutations of the scheduled tasks while limiting the number of generated permutations by exploiting temporal constraints. Note that this way, we can check the availability of a task faster, by interrupting the recursion as soon as we find a feasible permutation.

Our transition model allows adding tasks in any order by delegating part of the optimization—ordering and time setting—to the environment. Crucially, for any schedule s reached through this transition model, the execution times are optimal given the tasks scheduled in s . (We can also limit the number of generated permutations to lower the computation time, although we then lose the optimality guarantee.) This opens new possibilities for the choice of the decision algorithm, whose role is to solve the hardest part of the problem: inclusion.

3 Methods

The first decision algorithm we propose is a heuristic that we call highest cost-length ratio first (HCLR), which is similar to some knapsack problem heuristics. It consists in sorting tasks by their ratio between drop cost and length, then scheduling them in decreasing order (if they are still available when their turn comes). Equivalently, at each step, we choose the following action:

$$a = \operatorname{argmax}_{a \in A} \frac{C_{dr}^i}{L^i}$$

This heuristic empirically performs better than scheduling the task with the highest drop cost first, because it accounts for situations where for example two shorter, lower-priority tasks have a higher total drop cost than one longer, higher-priority task which they are incompatible with. A similar criterion has been proposed for radar resource management in Qu et al. [8]; however, the authors used the ratio in a task selection phase that preceded the scheduling itself, which is based on EST. Using our transition model, we can instead directly select and schedule each task in turn.

Our next decision algorithm is a version of Monte Carlo Tree Search (MCTS) adapted to task scheduling, which is based in large part on the version of Gaafer et al. [4]. MCTS uses a search tree where nodes represent MDP states and branches correspond to actions taken in the parent node [1]. The tree is constructed by successive rollouts: starting from the root node, which corresponds to the initial state, we select an action, apply the transition function to get the next node (which is created if needed), and repeat. Once a terminal state is reached, we can compute its total cost C , then backpropagate C up the path we just followed to update the best cost reached from each state-action pair: $C(s, a) \leftarrow \min\{C(s, a), C\}$. This value is then used in the computation of the upper-confidence bound $U(s, a)$ which determines which actions are chosen in the selection phase: $a = \operatorname{argmax}_{a'} U(s, a')$ with $U(s, a) = \frac{P(s, a)}{C(s, a)^\tau (1 + N(s, a))}$ where τ is a temperature, $N(s, a)$ is the number of rollouts where action a was taken in state s , and $P(s, a)$ is a prior probability function over actions. This selection rule is designed to balance exploration of the search tree and exploitation around the best solutions found so far. The prior, especially, plays a prominent role in steering exploration; a simple way to parameterize it is to sort the m available tasks according to a criterion, then assign them a respective prior probability of $p(1-p)^m$ for m ranging from 0 (for the first task) to $m-1$. For our experiments, we set $\tau = 2$ and $p = 0.6$, similarly to [4], but we diverge by using HCLR as our sorting criterion instead of EST.

One issue is that when using our transition model, since tasks can be scheduled in any order, it is possible to reach the same terminal state via multiple different rollouts. In order to prevent this, we structure our search tree similarly to a branch-and-bound (B&B) tree [2]: when a new action is taken in node s , leading to a new node s' , all actions explored in s in previous rollouts are made unavailable in s' and its descendants. This makes sure there is only one path

to each terminal state in the search tree. However, it also means that we may reach terminal nodes that are not complete schedules, when some tasks can still be scheduled, but have all been made unavailable by the previous rule. To limit the number of such situations, at each visit of a node s , we check if there exists a terminal node s' descended from s such that all tasks that are unexplored in s are scheduled in s' ; if so, these tasks are made unavailable in s . Additionally, to avoid performing the same rollout twice, if a node has no more available actions, the action that led to it is also made unavailable in the parent node, as in [4]. We call this algorithm B&B-MCTS.

Usually, in MCTS, when we reach a new state, we want to know which actions are available. With B&B-MCTS, this allows making the most effective use of pruning, according to the above rules. However, with our transition model, it requires partially solving (3) for each a priori available task to check if it has to be dropped. In larger instances, this involves significant computation, which reduces the number of rollouts that can be carried out in a given time. For this reason, we program B&B-MCTS so that it attempts to schedule tasks without prior verification, and drops them only if the attempt fails.

4 Results

To enable comparisons, we run experiments on instances from the same distribution as in [4]: $L^i \sim U(2, 15)$, $T_s^i \sim U(0, T_{max} - 12)$, $T_{du}^i = 0$, $T_{dr}^i - T_s^i \sim U(2, 12)$, $C_{dr}^i \sim U(100, 500)$, $C_{de}^i \sim U(1, 15)$, $T_{max} = 100$. The difficulty of an instance mostly depends on the density of tasks; by keeping T_{max} fixed, the difficulty can be controlled by setting the number of tasks.

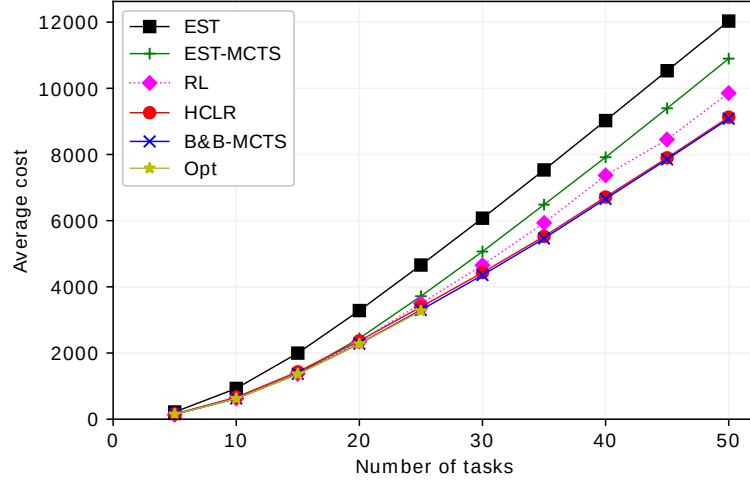
We compare our two methods, HCLR and B&B-MCTS, with the EST heuristic and with the version of MCTS proposed in [4] (which we term EST-MCTS). The run time of both versions of MCTS is limited to 1 second. All these algorithms are implemented in Python. We compute the optimal solution using MIP on instances where this resolution can be performed in a reasonable amount of time. The MIP solver we use is CBC [3]. These five algorithms are run on the same instances, 1000 per number of tasks. We also compare our results to those reported by [4] for a reinforcement learning-based extension of EST-MCTS inspired by AlphaZero [9].

Our results show that our approach provides a significant performance improvement over EST-MCTS, and even surpasses the reinforcement learning algorithm of [4], which used deep learning in conjunction with MCTS. Strikingly, the performance gain of B&B-MCTS over HCLR proves minimal. Moreover, with our implementation, HCLR's run time averages 36 milliseconds on 50-task instances, which would presumably make it more suitable for radar use cases than MCTS-based methods.

Unlike in [4], our approach allows due times distinct from start times, thus reflecting radar requirements more closely. We also run experiments on a similar task distribution but with $T_{du}^i \sim U(1, 4)$ and $T_{dr}^i - T_s^i \sim U(1, 8)$; our results match those of the first distribution very closely.

Table 1. Detailed statistics on three different instance types.

Algorithm	Avg. cost	Cost std. dev.	Dropped tasks (%)	Optimal solutions (%)	Avg. runtime (milliseconds)	Avg. number of rollouts
Number of tasks = 25, identical start and due times						
EST	4658.69	634.65	59.06	0.0	1.93	nan
EST-MCTS	3716.55	506.0	52.15	0.87	1002.88	727.06
HCLR	3390.6	551.68	48.2	21.35	13.12	nan
B&B-MCTS	3299.39	526.16	48.44	57.73	975.32	96.12
MIP-optimal	3268.43	523.85	48.65	100.0	1730.82	nan
Number of tasks = 50, identical start and due times						
EST	12031.8	857.18	77.64	nan	2.32	nan
EST-MCTS	10897.03	758.17	73.23	nan	1004.79	587.63
HCLR	9130.98	799.9	63.62	nan	36.7	nan
B&B-MCTS	9080.49	780.86	63.79	nan	991.72	30.1
Number of tasks = 25, distinct start and due times						
HCLR	3301.15	550.93	47.53	22.0	19.98	nan
B&B-MCTS	3219.61	524.51	47.61	56.5	994.69	70.96
MIP-optimal	3185.86	520.0	47.88	100.0	2219.44	nan

**Fig. 1.** Cost plotted against instance size for identical start and due times.

5 Conclusion

Through a reformulation of the problem of radar task scheduling, we were able to implement an efficient framework whose algorithmic applications can return quasi-optimal solutions in a limited amount of time. We see potential improvements for this framework, both by optimizing its run time to make it usable by real-world radars, and by increasing its flexibility; for example, being able to remove a task from a partial schedule could allow exploring the solution space more effectively. Furthermore, our B&B-MCTS algorithm could be extended with an AlphaZero-style reinforcement learning procedure which would make it able to adapt to various task distributions.

References

1. Browne, C.B., et al.: A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games* **4**(1), 1–43 (Mar 2012)
2. Clausen, J.: Branch and Bound Algorithms - Principles and Examples. Tech. rep., University of Copenhagen (2003)
3. COIN-OR Foundation: Cbc (COIN-OR Branch-and-Cut solver), <https://github.com/coin-or/Cbc>
4. Gaafar, M., et al.: Reinforcement Learning for Cognitive Radar Task Scheduling. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers (Nov 2019)
5. Jeauneau, V., Guenais, T., Barbaresco, F.: Scheduling on a fixed multifunction radar antenna with hard time constraint. *14th International Radar Symposium (IRS)* **1**, 375–380 (2013)
6. Laterre, A., et al.: Ranked Reward: Enabling Self-Play Reinforcement Learning for Combinatorial Optimization. *ArXiv* (2018)
7. Moo, P.W., Ding, Z.: Adaptive Radar Resource Management. Elsevier (2015)
8. Qu, Z., Ding, Z., Moo, P.: A Machine Learning Task Selection Method for Radar Resource Management (Poster). In: 2019 22th International Conference on Information Fusion (FUSION). pp. 1–6 (Jul 2019)
9. Silver, D., et al.: Mastering the game of Go without human knowledge. *Nature* **550**(7676), 354–359 (Oct 2017)
10. Winter, É., Baptiste, P.: On scheduling a multifunction radar. *Aerospace Science and Technology* **11**, 289–294 (2007)
11. Zheng, Q., Barbaresco, F., P.Baptiste: On scheduling a multifunction radar with duty cycle budget. *Cognitive Systems with Interactive Sensors* pp. 1–6 (2009)

Évaluation statistique efficace de la robustesse de classifieurs [★]

Karim TIT^{1,2}, Teddy Furon¹, Mathias Rousset¹, and Louis-Marie Traonouez²

¹ INRIA/IRISA, LinkMedia & SimSmart Teams, Rennes, France
`{karim.tit,teddy.furon,mathias.rousset}@inria.fr`

² Thales Land & Air Systems, La Ruche, Rennes, France
`{karim.tit,louis-marie.traonouez}@thalesgroup.com`

Abstract. Nous proposons de quantifier la robustesse d'un classifieur aux incertitudes d'entrée avec une simulation stochastique. L'évaluation de la robustesse est présentée comme un test d'hypothèse : le classifieur est considéré comme localement robuste si la probabilité de défaillance estimée est inférieure à un niveau critique. La procédure est basée sur une simulation d'Importance Splitting générant des échantillons d'événements rares. Nous dérivons des garanties théoriques non-asymptotiques par rapport à la taille de l'échantillon. Des expériences portant sur des classifieurs à grande échelle mettent en évidence l'efficacité de notre méthode.

Keywords: Apprentissage profond · Robustesse · Monte Carlo séquentiel

1 Introduction

Malgré des performances de pointe dans de nombreuses tâches de vision par ordinateur et de traitement automatique des langues, les réseaux neuronaux profonds (DNN) se sont révélés sensibles aux perturbations aléatoires et adverses [6,5].

Certification et évaluation de robustesse. La certification a posteriori vérifie le comportement correct d'un réseau entraîné $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. La propriété attendue est généralement définie localement : le réseau fonctionne correctement dans le voisinage $\mathcal{V}(\mathbf{x}_o) \subset \mathbb{R}^n$ d'une entrée particulière $\mathbf{x}_o \in \mathbb{R}^n$. En classification, la propriété prend le nom de *robustesse* et se lit comme suit : la sortie du réseau reste inchangée sur le voisinage $\mathcal{V}(\mathbf{x}_o)$. Cela certifie que le réseau est robuste aux incertitudes de support limité ou des perturbations adverses de distorsion contrainte. Le mécanisme de certification présente deux caractéristiques :

- Consistance : il ne certifie pas le réseau lorsque la propriété ne tient pas.
- Complétude : il certifie toujours le réseau lorsque la propriété est vérifiée.

Robustesse de corruption. La robustesse adversariale correspond à une analyse au pire cas, tandis que la *robustesse aux incertitudes* considère les perturbations aléatoires des entrées. L'ingrédient clé est l'introduction d'un

[★] Thèse financée par l'Agence de l'Innovation de Défense et Thales

modèle statistique π_0 des incertitudes épistémiques survenant le long de la chaîne d’acquisition de l’entrée. Par exemple, [5] prend des distributions Gaussiennes ou uniformes sur la boule $\mathcal{B}_{p,\epsilon}(\mathbf{x}_o)$ de rayon ϵ en norme ℓ_p centrée sur \mathbf{x}_o . Ils obtiennent des limites précises pour les classificateurs linéaires qu’ils étendent aux classificateurs non linéaires tels que des réseaux neuronaux profonds en supposant leurs "frontières de décision localement approximativement plates". L’approche présentée ici ne nécessite pas une telle hypothèse sur les classificateurs DNN.

Cette tendance récente s’accompagne d’une évaluation *quantitative* déterminant dans quelle mesure une propriété donnée est ou n’est pas présente. Par exemple, [13] estime la probabilité p qu’une propriété soit violée sous un modèle statistique donné des entrées. Cette approche n’a besoin d’aucune hypothèse sur le réseau examiné car il est utilisé comme une boîte noire. Elle peut donc s’appliquer aux réseaux profonds. La principale difficulté réside dans l’efficacité, c’est-à-dire la puissance de calcul nécessaire pour estimer les probabilités faibles. Leur manque de solidité provient de l’incapacité à déterminer si la probabilité p de violation est exactement nulle ou trop faible pour être estimée.

La section 2 présente un bref aperçu des procédures de certification en soulignant les hypothèses faites sur le réseau et leurs limites.

Ce travail présente une procédure efficace et passant à l’échelle pour évaluer la robustesse à la corruption sous un large panel de modèles statistiques. Il fournit une complétude et des garanties théoriques sur le manque de consistance.

2 Etat de l’art en matière d’évaluation de robustesse

Évaluation par l’exemple. Les attaques adverses avec contrainte de distorsion, comme l’attaque de descente de gradient projeté (PGD)[3], recherchent des violations de propriétés, c’est-à-dire des exemples adverses, à l’intérieur de la boule $\mathcal{B}_{p,\epsilon}(\mathbf{x}_o)$. Elles tirent parti du calcul rapide du gradient de la fonction du réseau grâce à la rétro-propagation. Ils sont rapides, mais ni consistants ni complets. Le réseau n’est pas certifié si l’attaque réussit, mais un échec ne dit rien sur la propriété : les attaques sont des processus empiriques sans garantie.

Certification formelle. En utilisant un solveur SMT (*Satisfiability Modulo Theories*), ReLUplex [8] fournit une méthode de certification consistante et complète, conçue pour les réseaux neuronaux avec des fonctions d’activation ReLU. Cependant, le même article montre que le problème de la certification consistante et complète des réseaux neuronaux (même restreint aux activations ReLU) est NP-complet. Le passage à l’échelle des grands réseaux modernes semble difficile. De plus, bien que ces méthodes formelles soient complètes en théorie, dans la pratique, la procédure peut abandonner ou se terminer de manière indéfinie avec un ‘timeout’ si le solveur sous-jacent est trop lent.

Certification incomplète. Pour passer à l’échelle, certains proposent des méthodes de vérification solides mais incomplètes par conception, en recourant à des approximations convexes. Le papier [12] obtient une accélération significative par rapport à ReLUplex et à d’autres certificateurs complets. Il introduit un benchmark de vérification appelé ERAN (voir Sect. 4). [15] présente une autre

certification incomplète basée sur des bornes fonctionnelles linéaires inférieures et supérieures de perceptrons multicouches (MLP) avec activation ReLU. Elle est généralisée aux MLP avec une fonction d'activation quelconque dans [16] et aux réseaux de neurones à convolution (CNNs) dans [2,14].

Ces méthodes de certification reposent sur des bornes inférieures de la distance minimale des exemples adverses. Elles sont donc pessimistes dans le sens où elles peuvent rejeter de nombreuses propriétés valides car la borne inférieure n'est pas toujours assez fine. Afin d'être "plus complet", [11] unifie ces méthodes de relaxation dans un cadre général et résout exactement la relaxation convexe optimale (pour des problèmes spécifiques sur CIFAR10 et MNIST) avec des ressources de calcul importantes. Les auteurs ont noté qu'une légère amélioration de la précision des bornes inférieures par rapport à l'état de l'art, ce qui suggère que cette approche a atteint sa limite.

Évaluation statistique. La robustesse à la corruption suppose un modèle statistique π_0 des entrées comme les distributions Gaussiennes ou uniformes sur la boule $\mathcal{B}_{p,\epsilon}(\mathbf{x}_o)$. Le papier [5] étudie la robustesse des réseaux de neurones linéaires et profonds. Ils obtiennent des limites précises pour les classifieurs linéaires qu'ils étendent aux classifieurs non linéaires avec des "frontières de décision localement approximativement plates". Le travail [13] définit la robustesse par la probabilité de défaillance suivante (meilleure d'autant que cette valeur est petite) :

$$p := \pi_0(\iota(\mathbf{X}|\mathbf{x}_o) = 1) = \int_{\mathbb{R}^n} \iota(\mathbf{x}|\mathbf{x}_o) \pi_0(d\mathbf{x}), \quad (1)$$

où $\iota(\cdot|\mathbf{x}_o)$ est la fonction indicatrice d'une défaillance. Cette évaluation statistique contraste fortement avec la littérature sur la robustesse adversariale qui adopte une analyse au pire cas. Un lien est établi lorsque π_0 est la distribution uniforme sur la boule $\mathcal{B}_{p,\epsilon}(\mathbf{x}_o)$: le volume de l'ensemble des exemples adverses est égal à $p \cdot \text{vol}(\mathcal{B}_{p,\epsilon}(\mathbf{x}_o))$. Cette probabilité p est parfois appelée la "densité adverse" [1].

La principale difficulté réside dans l'estimation de cette intégrale, en particulier lorsque l'événement $\{\iota(\mathbf{X}|\mathbf{x}_o) = 1\}$ est rare sous la distribution π_0 . Le papier [1] utilise une simulation de Monte Carlo peu efficace, [13] le fractionnement multi-niveaux (en anglais *multi-level splitting*) avec un mécanisme de rajeunissement basé sur l'algorithme de Metropolis-Hastings. Ces deux derniers travaux ne font aucune hypothèse sur le réseau car leurs procédures l'utilisent comme une boîte noire. Cela garantit le passage à l'échelle (dans le sens où elle s'applique aux réseaux profonds). L'efficacité du test statistique est mesurée par le temps d'exécution ou le nombre d'appels à la boîte noire.

L'évaluation quantitative revient à la certification en prenant une décision finale : le réseau est *réputé* correct si la probabilité de violation est inférieure à $p_c > 0$, une probabilité critique fixée par l'utilisateur.

3 L'évaluation de robustesse comme un test d'hypothèse

Notre approche utilise le test d'hypothèse statistique comme un ersatz de la certification. Comme dans [1], l'utilisateur fixe une faible probabilité critique p_c

et le test évalue si la probabilité de défaillance p est inférieure ou supérieure. Nous utilisons à ce moyen la simulation de la "dernière particule". Cette simulation dite de la "dernière particule" a été inventée par A. Guyader *et al.* [7]. C'est une variante efficace de l'échantillonnage multi-niveaux adaptatif employé par [13]. Nous montrons qu'avec une condition de terminaison bien choisie, cet algorithme est avantageux à la fois en termes d'efficacité et de garanties théoriques.

La section 3.1 présente l'algorithme "dernière particule" (*cf.* Alg. 1 en appendice), un classique dans le domaine de la simulation d'événements rares. La section 3.2 fait le lien entre la certification et les tests d'hypothèses statistiques dans le cadre de l'évaluation de la robustesse.

3.1 La simulation de la "dernière particule"

L'objectif de la simulation de la "dernière particule" est de générer efficacement des échantillons tirés aléatoirement selon une distribution de référence π_0 mais dans une région $\mathcal{R} := \{\mathbf{y} : h(\mathbf{y}) > 0\} \subset \mathbb{R}^n$, où $h : \mathbb{R}^n \rightarrow \mathbb{R}$, est la fonction dite *score*. Son efficacité est la capacité d'effectuer cette tâche en utilisant peu d'appels à la fonction de score, même lorsque la probabilité $\pi_0(\mathcal{R})$ est faible.

La simulation gère un ensemble de N particules (*i.e.* échantillons) qui sont i.i.d. par rapport à π_0 . Le nom "dernière particule" vient du fait que la simulation 'élimine' l'échantillon dont le score est le plus bas. Son score donne la valeur du niveau intermédiaire L_k à l'itération k (Alg. 1, ligne 6). Ensuite, cette particule est rafraîchie par échantillonnage selon π_0 mais conditionnée à l'événement $\{h(\mathbf{X}) > L_k\}$. Cette procédure d'échantillonnage est effectuée par la procédure $\text{Gen}(L_k, 1)$ (Alg. 1, ligne 11) détaillée dans l'Alg. 2. $\text{Gen}(-\infty, N)$ signifie alors simplement échantillonner N vecteurs aléatoires selon π_0 (ligne 3).

3.2 Lien avec la certification

Le test évalue si la probabilité de défaillance p est inférieure ou supérieure à une valeur critique p_c . Cela se fait en constatant que la simulation a atteint un nombre maximum d'itérations m (détaillé ci-après). L'algorithme s'arrête lorsque

- le nombre k d'itérations atteint l'entier m . Alors le réseau est certifié car on pense que $p < p_c$.
- le seuil intermédiaire $L_k > 0$ pour une itération $k < m$. Cela signifie que la simulation a généré quelques échantillons provoquant une défaillance. Alors le réseau n'est pas certifié.

Une certification peut être erronée pour deux raisons:

- La probabilité de défaillance est trop faible telle que $0 < p < p_c$. Cet écueil est évité en prenant une probabilité critique plus faible, mais cela est plus coûteux en temps de simulation.
- La probabilité de défaillance $p > p_c$, et un tel cas aurait du conduire à $L_k > 0$ pour un certain $k < m$. Mais comme la simulation est aléatoire, il y a une probabilité α que cette erreur se réalise.

On peut montrer que la relation entre le nombre d'itérations maximum m et le cahier des charges donné par les paramètres (p_c, α) est tel que :

Le quantile associé à la probabilité α pour la distribution $\Gamma(m, N)$ égale à $-\log p_c$.

Ainsi, l'entier m est une fonction décroissante de α pour p_c fixée, et évolue en $O(\log 1/p_c)$ pour α donné. Autrement dit, passer de $p_c = 10^{-30}$ à 10^{-60} multiplie m par deux et ainsi le temps de simulation. On voit ainsi clairement l'avantage de "dernière particule" par rapport à une simulation Monte Carlo.

4 Investigation expérimentale

Pour évaluer notre approche nous étudions un dispositif breveté [4] de surveillance aérienne qui permet l'identification du type d'aéronef à partir de données cinématiques des trajectoires. Ce dispositif s'intègre à des systèmes de contrôle qui collectent les informations issues de capteurs afin de suivre les trajectoires (modules de *tracking*) et de classer le type des trajectoires.

Dans cette étude nous analysons un jeu de données de trajectoires issues de capteurs ADSB (*Automatic Dependent Surveillance Broadcast*). Ce système équipe une majorité du trafic aérien, dont la totalité des vols commerciaux. Contrairement aux radars, c'est un système passif qui se repose sur le transpondeur interne aux aéronefs qui émet en continue les informations (non chiffrées) sur la position et l'identification de l'aéronef. Ce système permet de constituer facilement une base de données 'labelisée' pour entraîner et évaluer le dispositif de classification. Par ailleurs, l'intégration de ces données ouvertes aux systèmes de contrôle militaires est de plus en plus étudiée, mais il est préalablement nécessaire d'en vérifier la cohérence. Ce dispositif de classification pourrait par exemple détecter un aéronef qui falsifierait son identification.

Le dispositif d'apprentissage automatique étudié améliore la capacité de classification du type d'aéronef des systèmes de contrôle en calculant des caractéristiques cinématiques de la trajectoire. Pour chaque point sur la trajectoire, on calcule 9 caractéristiques, dont des mesures de vitesse, d'accélération, de courbure et de torsion en 2 ou 3 dimensions. Deux approches sont alors utilisées pour effectuer une classification des trajectoires. La première consiste à calculer pour chaque trajectoire et pour chacune des caractéristiques des mesures statistiques sur les valeurs des séries temporelles (valeurs minimales, maximales, 4 premier moments, quantiles). La seconde analyse les trajectoires entières de longueur variable à l'aide de réseaux de neurones récurrents.

Le jeu de données utilisé dans ce papier contient 26609 trajectoires de longueurs variables (de 26 à 1610 points). Ces trajectoires sont réparties en 6 classes selon le type d'aéronef (avion ou hélicoptère), la taille, le nombre et le type de moteur. La classe majoritaire contient 21529 trajectoires, la classe minoritaire 169. Le jeu de données est partitionné en un jeu d'entraînement de 21287 trajectoires et un jeu de validation de 5322 trajectoires.

Les expériences illustrent l'algorithme de la dernière particule présenté en section 3 d'une part et le système de certification formelle ERAN avec la méthode

Table 1. Données ADSB statiques – Comparaison ERAN [DeepPoly], Last Particle [$N = 2, p_c = 10^{-10}, t = 40$] et Monte Carlo simple [$N = 10^6, p_c = 10^{-10}$]. Moyennes sur 11 modèles de réseaux de neurones. Temps de vérification pour 100 trajectoires.

ε	ERAN		Last Particle		Monte Carlo simple	
	Certifié (%)	temps (sec. \pm std)	Validé (%)	temps (sec. \pm std)	Validé (%)	temps (sec. \pm std)
0.0001	100	5.0 ± 5.0	100	5.26 ± 0.1	100	7.74 ± 0.13
0.0005	100	5.01 ± 5.07	100	5.26 ± 0.10	100	8.19 ± 0.3
0.001	99	5.03 ± 5.06	100	5.26 ± 0.08	100	8.14 ± 0.27
0.005	98	4.91 ± 5.1	99	5.28 ± 0.11	99.8	7.86 ± 0.17
0.01	95	4.97 ± 5.2	98	5.21 ± 0.10	99	8.0 ± 0.21
0.05	20	6.88 ± 7.8	61	4.82 ± 0.3	89	7.74 ± 0.28
0.1	0.05	6.95 ± 8.33	43	4.0 ± 0.5	64	8.12 ± 0.2

DeepPoly [12] d’autre part. Les expériences ont été réalisées avec une carte graphique NVIDIA V100 et un processeur Intel Xeon Processor E5-2698 v4.

4.1 Expériences sur données ADSB statiques

Dans ces expériences, on analyse le jeu de données au format tabulaire comportant 72 caractéristiques. On entraîne pour commencer 3 modèles de réseaux de neurones avec respectivement 1 couche dense de 100, 500 ou 1000 neurones, et 1 modèle avec 3 couches denses de 500, 100 et 50 neurones. On compare les taux de certification d’ERAN et les taux de validation de la méthode Last Particle et d’un algorithme Monte Carlo naïf (avec 10^6) dans le tableau 1. Sur ces modèles de petites tailles on voit que la méthode DeepPoly d’ERAN a des temps similaires à notre procédure. Cependant, la variance des temps en fonctions des modèles et des trajectoires est plus élevé pour ERAN. Par comparaison, il est possible de borner facilement à l’avance le nombre d’appels fait au classifieurs que l’algorithme Last Particle. Par ailleurs pour des valeurs élevées du paramètre ε on voit que le temps de vérification du système DeepPoly augmente alors que notre méthode a tendance à accélérer avec ε croissant. Enfin, notons que jusqu’à un certain niveau de distortion l’algorithme de la dernière particule et ERAN donne les mêmes résultats et que ceux-ci divergent seulement pour des valeurs élevées d’ ε . Cette divergence s’explique d’ailleurs par la différence d’objectif: ERAN ne certifie que s’il n’existe aucune violation d’une sous-région donnée, tandis que notre méthode doit simplement vérifier que la probabilité d’échec dans cette même sous-région est assez faible.

Un des avantages de la méthode Last Particle proposée dans ce papier est qu’elle s’applique en boîte noire, indifféremment du type de modèle pour peu que l’on puisse définir une fonction score continue. Nous pouvons ainsi l’appliquer sur des modèles d’ensemble d’arbres de décision tels que des Random Forest ou du Gradient Boosting. On voit que le temps de vérification augmente globalement

Table 2. Données ADSB statiques – Analyse de modèles Random Forest (RF) et Gradient Boosting (GB) avec Last Particle [$N = 2, p_c = 10^{-10}, t = 40$]. Moyenne pour $\varepsilon \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$.

Nb. estimateurs	RF		GB	
	Validé (%)	temps (sec. \pm std)	Validé (%)	temps (sec. \pm std)
100	87.6	7.38 ± 0.004	73.6	11.9 ± 0.08
500	89.6	15.43 ± 0.065	76.2	32.98 ± 0.35
1000	90.6	28.55 ± 0.10	78.5	25.42 ± 0.09

Table 3. Données ADSB statiques – Entraînement adverse et entraînement stochastique sur un réseau de neurone. Comparaison de ERAN [DeepPoly] et Last Particle [$N = 2, p_c = 10^{-10}, t = 40$]. Temps de vérification pour 100 trajectoires.

Modèle	ERAN	Last Particle	
	Certifié (%)	Validé (%)	faux positifs (%)
Sans adv. training	73.14	86.29	13.14
adv. training ($\varepsilon = 0.025$, norme: l_2)	73.14	87.0	13.85
adv training ($\varepsilon = 0.025$, norme: l_∞)	73.57	96.0	22.42
random training ($\varepsilon = 0.01$, norme: l_2)	72.71	91.42	18.71
random training ($\varepsilon = 0.01$, norme: l_∞)	72.71	91.14	18.42

avec le nombre d’estimateurs du modèle d’ensemble. Cette augmentation en temps est cependant moins que linéaire (e.g. pour les forêts aléatoires, avec 10 fois plus d’estimateurs, le temps de calcul est seulement quadruplé). Le tableau 2 présente les résultats de la méthode Last Particle sur 6 modèles d’ensemble de tailles croissantes.

Nous présentons une dernière expérience qui compare le taux de certification de modèles de réseaux de neurones entraînés à l’aide de techniques de ‘robustification’ telles que l’entraînement adverse [9] et l’entraînement adverse stochastique [10]. Le tableau 3 montre que la certification formelle est quasiment insensible à ces techniques, alors que notre procédure détecte une amélioration de la robustesse.

4.2 Expériences sur données ADSB dynamiques

Cette section présente les résultats d’expériences sur des modèles utilisant l’approche dynamique du dispositif de classification, qui consiste à analyser les séries temporelles des trajectoires. Les données utilisées sont donc des trajectoires de longueurs variables comportant à chaque instant 9 caractéristiques. On entraîne pour commencer un premier modèle comportant des couches de convolutions avec un total de 289030 neurones. Le tableau 4 présente les résultats de certification de ce modèle avec ERAN et Last Particle.

Table 4. Données ADSB dynamiques – Comparaison ERAN [DeepPoly] et Last Particle [$N = 2, p_c = 10^{-10}, t = 40$] pour un réseau de neurones convolutif profond. Temps de vérification pour 100 trajectoires.

ε	ERAN		Last Particle		
	Certifié (%)	temps (sec. \pm std)	Validé (%)	temps (sec. \pm std)	faux positifs (%)
0.01	100	1553.5 ± 2396	100	332 ± 22	0
0.05	100	10686 ± 10368	100	319 ± 26	0

Pour conclure, nous présentons dans le tableau 5 des résultats de certification de 3 modèles plus complexes (31k, 63k et 134k paramètres, resp.), utilisant notamment des neurones récurrents de type LSTM. Pour ces modèles nous n’avons pu appliquer que la méthode Last Particle proposée dans ce papier.

Table 5. Données ADSB dynamiques – Certification de réseaux de neurones récurrents avec Last Particle [$N = 2, p_c = 10^{-10}, t = 40$]. Temps de vérification pour 100 trajectoires.

ε	Modèle 1		Modèle 2		Modèle 3	
	Validé (%)	temps (sec.)	Validé (%)	temps (sec.)	Validé (%)	temps (sec.)
0.01	100	639.4	100	5883	98.0	1460
0.05	97	562	100	5352	92.0	1330
0.1	90	619	100	6372	85.0	1292
0.5	27	311	13	3069	14.0	712

5 Conclusion

L’article propose une simulation stochastique pour évaluer la robustesse de modèles. Il prend les points de vue du test d’hypothèse (faux positif/faux négatif) et de la certification (complétude/consistance). La procédure proposée est efficace, complète et s’accompagne de garanties théoriques. Elle est aussi générale, fonctionnant avec des classifieurs en ‘boîte noire’ qu’il s’agisse de réseaux de neurones ou des forêts aléatoires par exemple. La principale limitation est que la simulation de la dernière particule est séquentielle, ce qui n’est pas compatible avec le GPU. Cependant, notre implémentation permet de traiter plusieurs entrées en parallèle.

Nos futurs travaux concernent son accélération lorsque la procédure est appliquée à un réseau de neurones en particulier. En effet, la procédure utilise le réseau seulement en ‘boîte noire’ et n’exploite pas le gradient ∇f de la fonction réseau f pourtant facilement calculable grâce à la rétro-propagation.

References

1. Baluta, T., Chua, Z.L., Meel, K.S., Saxena, P.: Scalable quantitative verification for deep neural networks. In: Proc. of Int. Conf. on Software Engineering (2021)
2. Boopathy, A., Weng, T.W., Chen, P.Y., Liu, S., Daniel, L.: CNN-Cert: An efficient framework for certifying robustness of convolutional neural networks. In: AAAI (Jan 2019)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE Computer Society, Los Alamitos, CA, USA (may 2017). <https://doi.org/10.1109/SP.2017.49>, <https://doi.ieeecomputersociety.org/10.1109/SP.2017.49>
4. Chopin, P., Barbaresco, F., Jouaber, S.: Dispositif d'identification d'un type d'aéronef, procédé d'identification et programme d'ordinateur associés, Office Européen des Brevets, 20169176.3, 14 octobre 2020
5. Franceschi, J.Y., Fawzi, A., Fawzi, O.: Robustness of classifiers to uniform ℓ_p and gaussian noise. In: Storkey, A., Perez-Cruz, F. (eds.) Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 84, pp. 1280–1288. PMLR (09–11 Apr 2018), <http://proceedings.mlr.press/v84/franceschi18a.html>
6. Gilmer, J., Ford, N., Carlini, N., Cubuk, E.: Adversarial examples are a natural consequence of test error in noise. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2280–2289. PMLR (09–15 Jun 2019), <http://proceedings.mlr.press/v97/gilmer19a.html>
7. Guyader, A., Hengartner, N., Matzner-Løber, E.: Simulation and estimation of extreme quantiles and extreme probabilities. Applied Mathematics & Optimization **64**, 171–196 (10 2011). <https://doi.org/10.1007/s00245-011-9135-z>
8. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: Majumdar, R., Kunčák, V. (eds.) Computer Aided Verification. pp. 97–117. Springer International Publishing, Cham (2017), <https://arxiv.org/abs/1312.6199>
9. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. In: 5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings. OpenReview.net (2017), <https://openreview.net/forum?id=BJm4T4Kgx>
10. Pinot, R., Meunier, L., Araujo, A., Kashima, H., Yger, F., Gouy-Pailler, C., Atif, J.: Theoretical evidence for adversarial robustness through randomization. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada. pp. 11838–11848 (2019), <https://proceedings.neurips.cc/paper/2019/hash/36ab62655fa81ce8735ce7cfdaf7c9e8-Abstract.html>
11. Salman, H., Yang, G., Zhang, H., Hsieh, C.J., Zhang, P.: A convex relaxation barrier to tight robustness verification of neural networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), <https://proceedings.neurips.cc/paper/2019/file/246a3c5544feb054f3ea718f61adfa16-Paper.pdf>
12. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. Proc. ACM Program. Lang. **3**(POPL) (Jan 2019). <https://doi.org/10.1145/3290354>, <https://doi.org/10.1145/3290354>

13. Webb, S., Rainforth, T., Teh, Y.W., Kumar, M.P.: A statistical approach to assessing neural network robustness. In: International Conference on Learning Representations (2019)
14. Weng, L., Chen, P.Y., Nguyen, L., Squillante, M., Boopathy, A., Oseledets, I., Daniel, L.: PROVEN: Verifying robustness of neural networks with a probabilistic approach. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6727–6736. PMLR (09–15 Jun 2019), <http://proceedings.mlr.press/v97/weng19a.html>
15. Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.J., Daniel, L., Boning, D., Dhillon, I.: Towards fast computation of certified robustness for ReLU networks. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 5276–5285. PMLR (10–15 Jul 2018), <http://proceedings.mlr.press/v80/weng18a.html>
16. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient neural network robustness certification with general activation functions. In: Advances in Neural Information Processing Systems (NeurIPS) (dec 2018)

A Pseudo-codes des algorithmes utilisés

Dans l'algorithme 1 ci-dessous, $\text{Comp_m}(p_c, \alpha, N)$ est une approximation numérique de plus petit entier tel que $P_{X \sim \Gamma(m, N)}[X \leq -\log(p_c)] = \alpha$.

Algorithm 1 Évaluation de robustesse avec l'algorithme de la dernière particule

Require: Nombre de particules N , niveau critique de probabilité p_c , niveau de confiance α

Ensure: Cert

```

1: Initialize:  $p \leftarrow 1 - 1/N$ ,  $k \leftarrow 1$ , Cert  $\leftarrow False$ , Stop  $\leftarrow False$ 
2:  $m \leftarrow \text{Comp\_m}(p_c, \alpha, N)$ 
3:  $\{\mathbf{x}_i\}_{i=1}^N \leftarrow \text{Gen}(-\infty, N)$ 
4: while  $k \leq m$  & Stop = False do
5:    $i^* \leftarrow \arg \min_{i \in 1:N} h(\mathbf{x}_i)$ 
6:    $L_k \leftarrow h(\mathbf{x}_{i^*})$ 
7:   if  $L_k > 0$  then
8:     Stop  $\leftarrow True$ 
9:      $P_{est} \leftarrow p^{k-1}$ 
10:  end if
11:   $\mathbf{x}_{i^*} \leftarrow \text{Gen}(L_k, 1)$ 
12:   $k \leftarrow k + 1$ 
13: end while
14: if Stop = False then
15:   Cert  $\leftarrow True$ 
16:    $P_{est} \leftarrow p_c$ 
17: end if
18: return Cert,  $P_{est}$ 

```

Algorithm 2 Échantillonnage conditionnelle d'une particule $\text{Gen}(L, 1)$

Require: seuil limite L , ensemble fini \mathcal{X} de particules dont le score est plus grand que L

Ensure: nouvelle particule \mathbf{X}

```

 $\mathbf{X} \leftarrow \mathcal{U}(\mathcal{X})$  ▷ On tire uniformément une particule dans  $\mathcal{X}$ 
for  $k = 1 : t$  do
   $\mathbf{Z} \leftarrow K(\mathbf{X}, s)$  ▷ Transition  $\pi_0$ -réversible.
  if  $h(\mathbf{Z}) > L$  then ▷ Rejet
     $\mathbf{X} \leftarrow \mathbf{Z}$ 
  end if
end for
return  $\mathbf{X}$ 

```

Generalized $SU(1, 1)$ Equivariant Convolution on Fock-Bargmann Spaces for Robust Radar Doppler Signal Classification

Pierre-Yves Lagrave¹[0000–0002–5774–636X] and Frédéric Barbaresco²[0000–0003–3664–3609]

¹ Thales Research and Technology, Palaiseau, France

`pierre-yves.lagrange@thalesgroup.com`

² Thales Land and Air Systems, Meudon, France

`frederic.barbaresco@thalesgroup.com`

Abstract. Classifying radar Doppler signals with Deep Learning algorithms is a challenging task, in particular because of the noisy nature of the data (clutter, thermal noise, etc.). Equivariant Neural Networks (ENN) have already been shown very promising in this context by coupling hyperbolic embedding techniques with dedicated $SU(1, 1)$ convolution operators in order to achieve local robustness by-design. In this paper, we introduce a generalized $SU(1, 1)$ equivariant convolution operator on the Fock-Bargmann spaces by leveraging on the representations of $SU(1, 1)$ over these functional Hilbert spaces. We further give a new way of sampling over $SU(1, 1)$ for Monte-Carlo computations by using a generalization of the Bloch-Messiah decomposition of elements of the symplectic group $SL(2, \mathbb{R})$ to those of $SU(1, 1)$. We finally illustrate our approach on the problem of radar clutter classification and demonstrate in this context that $SU(1, 1)$ ENN achieve better performance results than conventional approaches from both accuracy and robustness standpoints.

Keywords: Equivariant convolution · Group representation · Monte-Carlo sampling · Radar clutter classification

1 Introduction

Recognizing radar Doppler signatures with Machine learning algorithms has been investigated by exploiting multiple representations of the radar signals, including the use of off-the-shelf algorithms such as Convolutional Neural Networks (CNN) to process Doppler spectrum images [20], the use of Complex-Valued Neural Networks (CVNN) [1], and the use of geometric approaches allowing to process Doppler signals represented by their complex covariance matrices [3, 5]. Building on the latter, it was recently shown that the robustness of Deep Learning algorithms could be improved for Doppler signal processing tasks by leveraging on Geometric Deep Learning methods [2].

Geometric Deep Learning is an emerging field getting more and more traction because of its successful application to a wide range of domains [11, 15, 8,

9]. In this context, Equivariant Neural Networks (ENN) [12] have been shown to be superior to conventional Deep Learning approaches from both accuracy and robustness standpoints and appear as a natural alternative to data augmentation techniques to achieve geometrical robustness with respect to semantically preserving transforms such as isometries. More precisely, ENN were initially introduced in [7] for image classification by leveraging on group-based equivariant convolution operators and are now achieving state-of-the-art accuracies for a wide range of applications, including for Computer Vision, Graph and Point Cloud processing, Simulation and Trajectory prediction, in Reinforcement Learning and for Time Series Analysis. Furthermore, ENN are also very appealing from a safety standpoint as achieving robustness-by-design, making them generally promising for Defense related applications [16].

Achieving equivariance with respect to the $SU(1, 1)$ group is of particular interest in the context of radar Doppler signal classification when representing the signals as complex covariance matrices [3, 5] and leveraging on hyperbolic embedding techniques to represent the input data as graphs of functionals defined on the Poincaré disk \mathbb{D} [18]. In particular, the authors have proposed in [17] using the equivariant convolution operator defined in [14] for functions $f : \mathbb{D} \rightarrow \mathbb{C}$ and to rely on a regular action of $SU(1, 1)$ on those functions. However, other group actions may need to be envisioned to better account for plausible real-world deformations of the original input data and to improve robustness accordingly, as shown in [18] with respect to thermal noise effects.

In this paper, we introduce a new $SU(1, 1)$ equivariant convolution operator by leveraging on Unitary Irreducible Representations (UIR) of $SU(1, 1)$ on the Fock-Bargmann Hilbert spaces, as described in [10]. Leveraging on recent results of [13] with respect to the extension of Bloch-Messiah decomposition of symplectic matrices to $SU(1, 1)$, we also propose an alternative sampling method to that used in [17] for computing Monte-Carlo estimations of $SU(1, 1)$ -based convolution operators. Finally, we illustrate the approach in the context of radar clutter classification by first working on data simulated according to a realistic model and then providing some preliminary results obtained on a real-world dataset.

2 Mathematical Background

We denote \mathbb{D} the Poincaré unit disk $\mathbb{D} = \{z = x + iy \in \mathbb{C} / |z| < 1\}$ and then consider the following Lie Group:

$$SU(1, 1) = \left\{ g_{\alpha, \beta} = \begin{bmatrix} \alpha & \beta \\ \bar{\beta} & \bar{\alpha} \end{bmatrix}, |\alpha|^2 - |\beta|^2 = 1, \alpha, \beta \in \mathbb{C} \right\} \quad (1)$$

We can endow \mathbb{D} with a transitive action \circ of $SU(1, 1)$ defined as it follows

$$\forall g_{\alpha, \beta} \in SU(1, 1), \forall z \in \mathbb{D}, g_{\alpha, \beta} \circ z = \frac{\alpha z + \beta}{\bar{\beta} z + \bar{\alpha}} \quad (2)$$

As highlighted in [10], this action can be extended to functions of the Fock-Bargmann Hilbert space \mathcal{FB}_η , for $\eta = 1, \frac{3}{2}, 2, \frac{5}{2}, \dots$, through the UIR represen-

tation ρ^η of $SU(1, 1)$ on \mathcal{FB}_η which is defined as it follows, for $f \in \mathcal{FB}_\eta$ and $z \in \mathbb{D}$:

$$[\rho^\eta(g_{\alpha,\beta})(f)](z) = \frac{1}{(\alpha - \bar{\beta}z)^{2\eta}} f\left(\frac{\bar{\alpha}z - \beta}{\alpha - \bar{\beta}z}\right) = \frac{1}{(\alpha - \bar{\beta}z)^{2\eta}} f(g_{\alpha,\beta}^{-1} \circ z) \quad (3)$$

Figure 1 illustrates this action of $SU(1, 1)$ and provides a comparison with the regular action considered in [17] and defined by $\rho^0(g_{\alpha,\beta})(f) = f(g_{\alpha,\beta}^{-1} \circ z)$, showing in particular that several perturbations can be captured through the representations ρ^η as η varies.

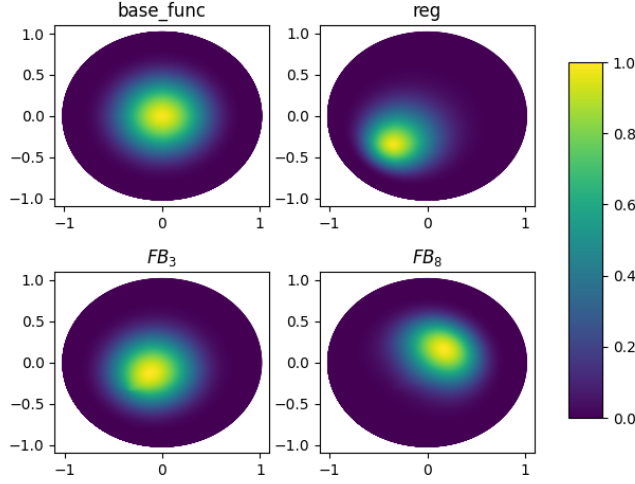


Fig. 1. Example of the action of $g_{\alpha,\beta} \in SU(1, 1)$ on the Fock-Bargmann Hilbert spaces \mathcal{FB}_η . From top to bottom and left to right: the original function f being a Gaussian kernel on \mathbb{D} , the transformed function $\rho^0(g_{\alpha,\beta})(f)$ by the regular action and the transformed functions $\rho^3(g_{\alpha,\beta})(f)$ and $\rho^8(g_{\alpha,\beta})(f)$.

3 Generalized Convolution

We generalize here the convolution operator considered in [17] and show that it allows achieving equivariance with respect to the action of $G = SU(1, 1)$ represented by ρ^η .

To do so, we define for $z \in \mathbb{D}$,

$$\psi_{f,k}^\eta(z) = \int_G [\rho^\eta(g)(k)](z) [\rho^\eta(g)^{-1}(f)](0_{\mathbb{D}}) d\mu^G(g) \quad (4)$$

with $0_{\mathbb{D}}$ the center of \mathbb{D} and where μ^G refers to the Haar measure of G that is normalized according to $\int_G F(g \circ 0_{\mathbb{D}}) d\mu^G(g) = \int_{\mathbb{D}} F(z) dm(z)$ for all $F \in L^1(\mathbb{D}, dm)$, and where the measure dm is given for $z = z_1 + iz_2$ by $dm(z) = \frac{dz_1 dz_2}{(1-|z|^2)^2}$.

Proposition 1. *The operator $f \rightarrow \psi_{f,k}^\eta$ is equivariant with respect to the action of $\text{SU}(1,1)$ represented by ρ^η , in the following sense,*

$$\forall g_{\alpha_0, \beta_0} \in \text{SU}(1,1), \quad \rho^\eta(g_{\alpha_0, \beta_0}) \left(\psi_{f,k}^\eta \right) = \psi_{\rho^\eta(g_{\alpha_0, \beta_0})(f), k}^\eta \quad (5)$$

Proof. $\forall g_0 = g_{\alpha_0, \beta_0} \in G, \forall z \in \mathbb{D}$, we have:

$$\rho^\eta(g_0) \left(\psi_{f,k}^\eta \right) (z) = \frac{1}{(\alpha_0 - \bar{\beta}_0 z)^{2\eta}} \int_G [\rho^\eta(g)(k)] (g_0^{-1} \circ z) \left[\rho^\eta(g)^{-1}(f) \right] (0_{\mathbb{D}}) d\mu^G(g)$$

$\forall g = g_{\alpha, \beta} \in G$, it is also possible to write

$$\rho^\eta(g)(k) (g_0^{-1} \circ z) = \frac{(\alpha_0 - \bar{\beta}_0 z)^{2\eta}}{(A - \bar{B}z)^{2\eta}} k \left((g_0 g)^{-1} \circ z \right)$$

with $A = \alpha\alpha_0 + \bar{\beta}\beta_0$ and $\bar{B} = \bar{\beta}\bar{\alpha}_0 + \alpha\bar{\beta}_0$, so that $\alpha = A\bar{\alpha}_0 - \beta_0\bar{B}$. We then have

$$\begin{aligned} \rho^\eta(g_0) \left(\psi_{f,k}^\eta \right) (z) &= \int_G \frac{1}{(A - \bar{B}z)^{2\eta}} k \left((g_0 g)^{-1} \circ z \right) \frac{1}{\bar{\alpha}^{2\eta}} f(g \circ 0_{\mathbb{D}}) d\mu^G(g) \\ &= \int_G \frac{1}{(A - \bar{B}z)^{2\eta}} k \left((g_0 g)^{-1} \circ z \right) \frac{1}{(\alpha_0 \bar{A} - \bar{B}\beta_0)^{2\eta}} f(g \circ 0_{\mathbb{D}}) d\mu^G(g) \\ &= \int_G \frac{1}{(A - \bar{B}z)^{2\eta}} k(\tilde{g}^{-1} \circ z) \frac{1}{(\alpha_0 \bar{A} - \bar{B}\beta_0)^{2\eta}} f \left((\tilde{g}^{-1} g_0)^{-1} \circ 0_{\mathbb{D}} \right) d\mu^G(\tilde{g}) \\ &= \int_G [\rho^\eta(\tilde{g})(k)](z) \left[\rho^\eta(\tilde{g})^{-1}(\rho^\eta(g_0)(f)) \right] (0_{\mathbb{D}}) d\mu^G(\tilde{g}) \\ &= \psi_{\rho^\eta(g_0)(f), k}^\eta(z) \end{aligned}$$

where we have used the change of variable $\tilde{g}_{A,B} = \tilde{g} = g_0 g$ and the invariance property of the Haar measure.

4 Numerical Computation

In order to numerically compute the convolution (4), we can use a Monte-Carlo technique following the approach introduced in [11] and then consider the following estimator

$$\psi_{f,k}^{\eta, N}(z) = \frac{1}{N} \sum_{i=1}^N [\rho^\eta(g_i)(k)](z) \left[\rho^\eta(g_i)^{-1}(f) \right] (0_{\mathbb{D}}) \quad (6)$$

where the samples g_i are drawn according to the Haar measure μ^G of G .

Motivated by the Cartan decomposition of G , [17] proposes sampling in $SU(1, 1)$ by first drawing elements in \mathbb{D} seen as the cosets space $SU(1, 1)/U(1)$ and then lifting to $SU(1, 1)$ by drawing random elements in the rotation group $U(1)$. We propose here an alternative approach by leveraging on the result of [13] with respect to the extension of the Bloch-Messiah decomposition of symplectic matrices to $SU(1, 1)$ elements.

More precisely, the group elements can actually be parameterized by two angles γ and γ' and one real parameter d , so that

$$\begin{aligned} g_{\rho, \gamma, \gamma'} &= \begin{bmatrix} e^{i\gamma} & 0 \\ 0 & e^{-i\gamma} \end{bmatrix} \begin{bmatrix} \cosh \rho & \sinh \rho \\ \sinh \rho & \cosh \rho \end{bmatrix} \begin{bmatrix} e^{i\gamma'} & 0 \\ 0 & e^{-i\gamma'} \end{bmatrix} \\ &= \begin{bmatrix} e^{i(\gamma+\gamma')} \cosh \rho & e^{i(\gamma-\gamma')} \sinh \rho \\ e^{-i(\gamma-\gamma')} \sinh \rho & e^{-i(\gamma+\gamma')} \cosh \rho \end{bmatrix} = g_{\alpha, \beta} \end{aligned}$$

with $\alpha = e^{i(\gamma+\gamma')} \cosh \rho$ and $\beta = e^{i(\gamma-\gamma')} \sinh \rho$. The following proposition gives the corresponding Haar measure that could then be used to sample elements $g_{\rho, \gamma, \gamma'} \in SU(1, 1)$ to compute the Monte-Carlo estimator (6).

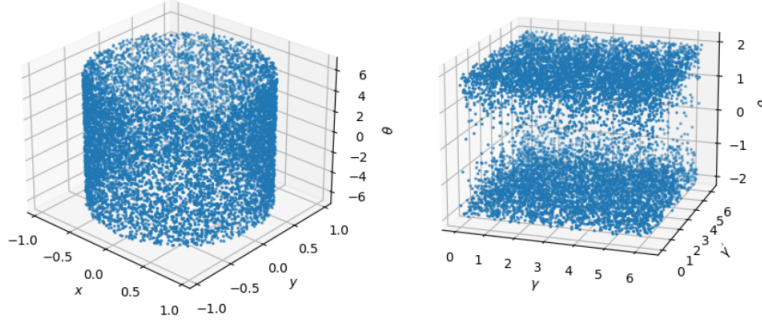


Fig. 2. Left: sampling according to the Cartan parameterization for which $SU(1, 1) \ni g_{\alpha, \beta} = g_{x, y, \theta}$, with $z = x + iy = \frac{\beta}{\alpha}$ and $\theta = 2 \arg \alpha$. Right: sampling according to the Bloch-Messiah parameterization for which $SU(1, 1) \ni g_{\alpha, \beta} = g_{d, \gamma, \gamma'}$, with $\alpha = e^{i(\gamma+\gamma')} \cosh \rho$ and $\beta = e^{i(\gamma-\gamma')} \sinh \rho$.

Proposition 2. *The normalized Haar measure corresponding to the Bloch-Messiah parameterization of $SU(1, 1)$ is given by*

$$d\mu^G(g_{\rho, \gamma, \gamma'}) = \frac{1}{4\pi^2} |\sinh 2\rho| d\rho d\gamma d\gamma' \quad (7)$$

Proof. Based on [6], we remind ourselves that an element $g_{\alpha,\beta} \in \text{SU}(1,1)$ can be written as $g_{\alpha,\beta} = t\mathbf{1} + iz\sigma_z + x\sigma_x + y\sigma_y$, with $\mathbf{1}$ the identity matrix and $\sigma_x, \sigma_y, \sigma_z$ the three Pauli matrices, where we have used the notations $\alpha = t + iz$ and $\beta = x - iy$ with $t^2 + z^2 - x^2 - y^2 = 1$. With such a parameterization, we can define the invariant Haar measure of the group by

$$d\mu^G(g_{\alpha,\beta}) = \frac{1}{\sqrt{1+x^2+y^2-z^2}} dx dy dz \quad (8)$$

If we consider the Bloch-Messiah parameterization for which $\alpha = e^{i(\gamma+\gamma')} \cosh \rho$ and $\beta = e^{i(\gamma-\gamma')} \sinh \rho$, we then have to consider the following change of variables, $x = \sinh \rho \cos(\gamma - \gamma')$, $y = -\sinh \rho \sin(\gamma - \gamma')$ and $z = \cosh \rho \sin(\gamma + \gamma')$, for which the absolute determinant of the Jacobian matrix is $2 \cosh^2 \rho |\sinh \rho \cos(\gamma + \gamma')|$. We then have

$$d\mu^G(g_{\rho,\gamma,\gamma'}) = \frac{2 \cosh^2 \rho |\sinh \rho \cos(\gamma + \gamma')|}{\sqrt{\cosh^2 \rho \cos(\gamma + \gamma')}} d\rho d\gamma d\gamma' = |\sinh 2\rho| d\rho d\gamma d\gamma' \quad (9)$$

The measure stated in (7) is then obtained after re-normalizing the above equality for the angular part.

Figure 2 illustrates the sampling of $\text{SU}(1,1)$ according to the Cartan (left) and Bloch-Messiah (right) parameterizations, the two pictures representing the same group elements but with different parameterizations. It is also interesting to notice that as the left-handside of Figure 2 can be folded along its θ axis, the Cartan parameterization actually corresponds to a torus with \mathbb{D} as orthogonal sections.

5 Application to Radar Clutter Classification

In the following, we focus on radar clutter classification and consider the setup introduced in [5], in which the signals are represented as Toeplitz Hermitian Positive Definite (THPD) covariance matrices of dimension n . A $\text{SU}(1,1)$ equivariant neural network can operate on the corresponding data by leveraging on the Trench-Verblunsky theorem allowing to identify n -dimensional THPD matrices with $n-1$ reflection coefficients $\mu_i \in \mathbb{D}$ after adequate rescaling. A lifting step as introduced in [17] is then used to represent a THPD matrix Γ as a complex signal f_Γ on the Poincaré disk \mathbb{D} .

More precisely, we represent each spatial cell by its THPD auto-correlation matrix Γ , our goal being to predict the corresponding clutter $c \in \{1, \dots, n_c\}$ from the observation of Γ . Within our formalism, the training samples are of the form (f_{Γ_i}, c_i) .

5.1 Results on Simulated Data

We have conducted some initial testing by simulating a given cell according to

$$Z = \sqrt{\tau} R^{1/2} x + b_{radar} \quad (10)$$

where τ is a positive random variable corresponding to the clutter texture, R a THPD matrix associated with a given clutter, $x \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^x)$ and $b_{radar} \sim \mathcal{N}_{\mathbb{C}}(0, \sigma)$, with $\mathcal{N}_{\mathbb{C}}(0, t)$ referring to the complex gaussian distribution with mean 0 and standard deviation t . In the following, b_{radar} will be considered as a source of thermal noise inherent to the sensor.

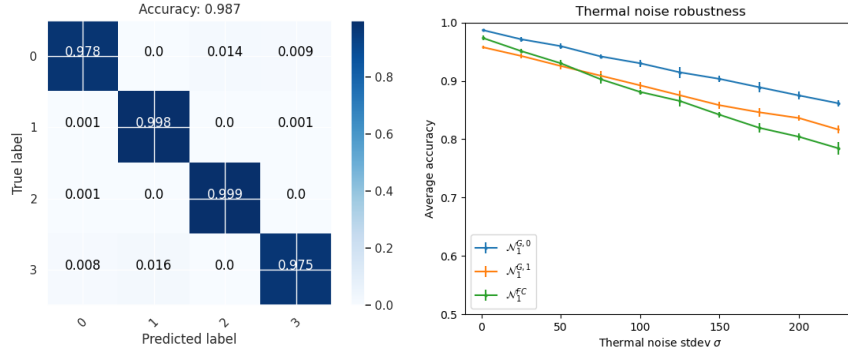


Fig. 3. Left handside: confusion matrix corresponding to the evaluation of $\mathcal{N}_1^{G,0}$ on the testing set T_1 , averaged over 10 realizations. Right handside: average accuracy results of the algorithms $\mathcal{N}_1^{G,0}$, $\mathcal{N}_1^{G,1}$ and \mathcal{N}_1^{FC} on the testing sets T_σ shown as a function of σ , together with the corresponding standard deviation as error bars

We have instantiated some simple neural networks constituted of one SU(1, 1) ρ^η -convolutional layer with two filters and ReLu activation functions, followed by one fully connected layer and one softmax layer operating on the complex numbers represented as 2-dimensional tensors. The kernel functions are modeled as some neural networks with one layer of 16 neurons with swish activation functions, combined with the Riemannian logarithm of \mathbb{D} . The two convolution maps have been evaluated on the same grid constituted of 100 elements of \mathbb{D} sampled according to the corresponding volume measure.

To appreciate the improvement provided by our approach, we will compare the obtained results with those corresponding to the use of a fully connected neural network with roughly the same number of trainable parameters and operating on the complex reflection coefficients. In the following, we will denote $\mathcal{N}_\sigma^{G,\eta}$ (resp. \mathcal{N}_σ^{FC}) the neural network with SU(1, 1) ρ^η -equivariant convolutional (resp. fully connected) layers and trained on 400 THPD matrices of dimension 10 corresponding to 4 different classes (100 samples in each class) which have been simulated according to (10) with a thermal noise standard deviation σ .

In order to evaluate the algorithms $\mathcal{N}_\sigma^{G,\eta}$ and \mathcal{N}_σ^{FC} , we have considered several testing sets T_σ consisting in 2000 THPD matrices of dimension 10 (500 samples in each of the 4 classes) simulated according to (10) with a thermal noise standard deviation σ . The obtained results are shown on Figure 3 where it can in particular be seen that \mathcal{N}_1^{FC} consistently achieves lower accuracy results than $\mathcal{N}_1^{G,0}$, hence demonstrating the superiority of our approach from both accuracy and robustness standpoints. We can also see that considering different representations ρ^η has an impact on the algorithm robustness as, although achieving slightly lower accuracy results in the standard regime ($\sigma = 1$), $\mathcal{N}_1^{G,1}$ eventually outperforms \mathcal{N}_1^{FC} as σ increases. It however reaches a lower robustness degree than $\mathcal{N}_1^{G,0}$ with respect to the considered perturbations.

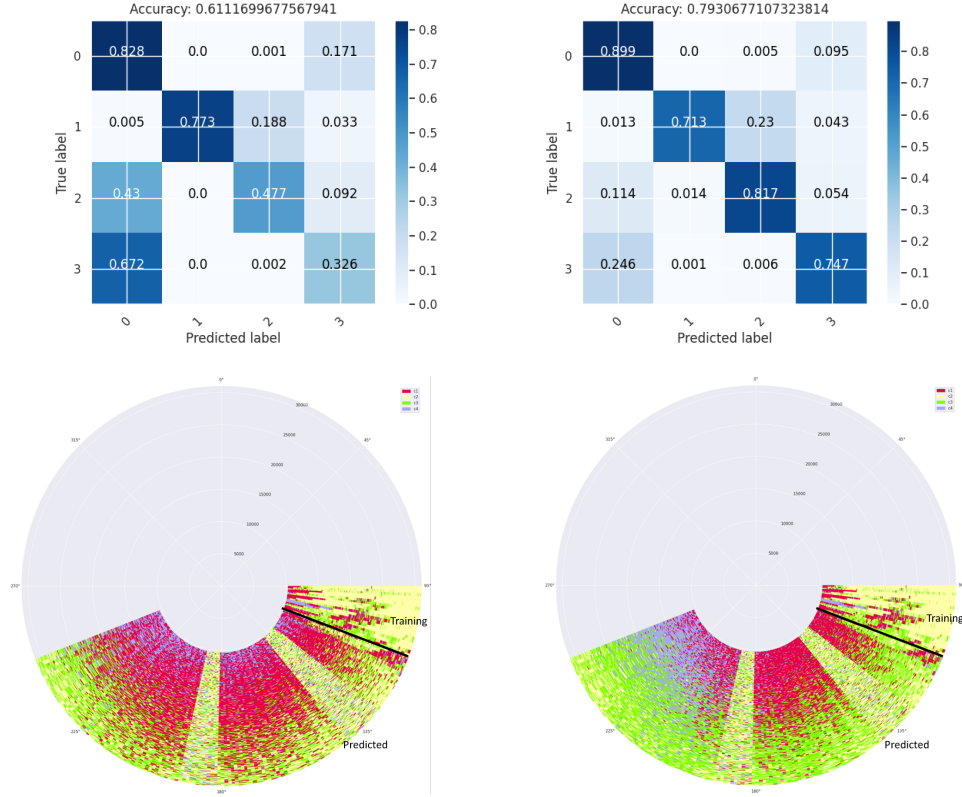


Fig. 4. Top: confusion matrices of \mathcal{N}_1^{FC} (left) and $\mathcal{N}_1^{G,0}$ (right) when trained on 400 cells randomly sampled in the training area. Bottom: predicted clutters from \mathcal{N}_1^{FC} (left) and $\mathcal{N}_1^{G,0}$ (right).

5.2 Preliminary Testing on Real Data

We have then conducted some preliminary testing on the real-world dataset recorded by a radar located near Saint-Mandrier (France) and which has been introduced and labelled in [4]. The results obtained with the ENN $\mathcal{N}_1^{G,0}$ (with kernel functions of 128 hidden neurons) and with a comparable fully connected architecture $\mathcal{N}_1^{\text{FC}}$ are shown on Figure 4, where we can in particular see that the results obtained on simulated data (Section 5.1) seem to be confirmed in the real world as $\mathcal{N}_1^{G,0}$ achieves almost 80% of accuracy while $\mathcal{N}_1^{\text{FC}}$ only reaches 61% in the considered set-up.

6 Conclusion and Further Work

Motivated by the successful application of SU(1, 1) ENN to Doppler signal classification [18], we have generalized the convolution operator considered in [17] in order to handle more general group actions through the representation of SU(1, 1) on the Fock-Bargmann spaces. We have shown that our generalized operator is equivariant with respect to the considered action of SU(1, 1), so that it could be used to build equivariant layers of ENN. We have then proposed a sampling method for computing convolution operators with Monte-Carlo estimators by leveraging on the Bloch-Messiah parameterization of SU(1, 1), an approach complementary to that relying on the Cartan decomposition. We have finally illustrated our approach on simulated clutter data and shown its superiority with respect conventional Deep Learning algorithms from both accuracy and robustness standpoints.

Further work will include the study of numerical methods other than Monte-Carlo approaches which suffer from scalability issues when the convolution operators are used within deep ENN architectures and establishing some links with the coadjoint representation theory [19] may be useful in this context. Also, by leveraging on the isomorphism between SU(1, 1) and $\text{SL}(2, \mathbb{R})$, we will investigate extending the approach presented in this paper to cover the action of $\text{SL}(2, \mathbb{R})$ on \mathbb{H}_2 and to build corresponding ENN in order to achieve robustness to a wider range of real-world perturbations.

References

1. Barrachina, J.A., Ren, C., Morisseau, C., Vieillard, G., Ovarlez, J.P.: Complex-valued vs. real-valued neural networks for classification perspectives: An example on non-circular data. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2990–2994 (2021). <https://doi.org/10.1109/ICASSP39728.2021.9413814>
2. Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges (2021)
3. Brooks, D., Schwander, O., Barbaresco, F., Schneider, J., Cord, M.: A hermitian positive definite neural network for micro-doppler complex covariance processing. In: 2019 International Radar Conference (RADAR). pp. 1–6 (2019). <https://doi.org/10.1109/RADAR41533.2019.171277>

4. Cabanes, Y., Barbaresco, F., Arnaudon, M., Bigot, J.: Unsupervised Machine Learning for Pathological Radar Clutter Clustering: the P-Mean-Shift Algorithm. Rennes, France (Nov 2019), <https://hal.archives-ouvertes.fr/hal-02875430>
5. Cabanes, Y., Barbaresco, F., Arnaudon, M., Bigot, J.: Toeplitz hermitian positive definite matrix machine learning based on fisher metric. In: Nielsen, F., Barbaresco, F. (eds.) *Geometric Science of Information*. pp. 261–270. Springer International Publishing, Cham (2019)
6. Chiribella, G., D’Ariano, G.M., Perinotti, P.: Applications of the group $su(1, 1)$ for quantum computation and tomography. *Laser Physics* **16**(11), 1572–1581 (Nov 2006). <https://doi.org/10.1134/s1054660x06110119>, <http://dx.doi.org/10.1134/S1054660X06110119>
7. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: Balcan, M.F., Weinberger, K.Q. (eds.) *Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 48, pp. 2990–2999. PMLR, New York, New York, USA (20–22 Jun 2016), <http://proceedings.mlr.press/v48/cohen16.html>
8. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical cnns. *CoRR abs/1801.10130* (2018), <http://arxiv.org/abs/1801.10130>
9. Cohen, T.S., Weiler, M., Kicanaoglu, B., Welling, M.: Gauge equivariant convolutional networks and the icosahedral cnn (2019)
10. del Olmo, M.A., Gazeau, J.P.: Covariant integral quantization of the unit disk. *Journal of Mathematical Physics* **61**(2), 022101 (Feb 2020). <https://doi.org/10.1063/1.5128066>
11. Finzi, M., Stanton, S., Izmailov, P., Wilson, A.G.: Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data (2020)
12. Gerken, J.E., Aronsson, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., Persson, D.: Geometric deep learning and equivariant neural networks (2021)
13. Grain, J., Vennin, V.: Canonical transformations and squeezing formalism in cosmology. *Journal of Cosmology and Astroparticle Physics* **2020**(02), 022–022 (Feb 2020). <https://doi.org/10.1088/1475-7516/2020/02/022>, <http://dx.doi.org/10.1088/1475-7516/2020/02/022>
14. Helgason, S.: *Groups and geometric analysis* (1984)
15. Lafarge, M.W., Bekkers, E.J., Pluim, J.P.W., Duits, R., Veta, M.: Roto-translation equivariant convolutional networks: Application to histopathology image analysis (2020)
16. Lagrave, P.Y., Barbaresco, F.: Introduction to Robust Machine Learning with Geometric Methods for Defense Applications (Jul 2021), <https://hal.archives-ouvertes.fr/hal-03309807>, working paper or preprint
17. Lagrave, P.Y., Cabanes, Y., Barbaresco, F.: "su(1,1) equivariant neural networks and application to robust toeplitz hermitian positive definite matrix classification". In: Nielsen, F., Barbaresco, F. (eds.) *Geometric Science of Information*. pp. 577–584. Springer International Publishing, Cham (2021)
18. Lagrave, P.Y., Cabanes, Y., Barbaresco, F.: An equivariant neural network with hyperbolic embedding for robust doppler signal classification. In: 2021 21st International Radar Symposium (IRS). pp. 1–9 (2021). <https://doi.org/10.23919/IRS51887.2021.9466226>
19. Rieffel, M.A.: Lie group convolution algebras as deformation quantizations of linear poisson structures. *American Journal of Mathematics* **112**(4), 657–685 (1990)
20. Trommel, R., Harmanny, R., Cifola, L., Driessen, J.: Multi-target human gait classification using deep convolutional neural networks on micro-doppler spectrograms. In: 2016 European Radar Conference (EuRAD). pp. 81–84 (2016)

Case-based reasoning for rare events prediction on strategic sites

Vincent Vidal, Marie-Caroline Corbineau, and Tugdual Ceillier

Preligens (ex-Earthcube), Paris, France
www.preligens.com, [name].[surname]@preligens.com

Abstract. Satellite imagery is now widely used in the defense sector for monitoring locations of interest. Although the increasing amount of data enables pattern identification and therefore prediction, carrying this task manually is hardly feasible. We hereby propose a case-based reasoning approach for automatic prediction of rare events on strategic sites. This method allows direct incorporation of expert knowledge, and is adapted to irregular time series and small-size datasets. Experiments are carried out on two use-cases using real satellite images: the prediction of submarines arrivals and departures from a naval base, and the forecasting of imminent rocket launches on two space bases. The proposed method significantly outperforms a random selection of reference cases on these challenging applications, showing its strong potential.

Keywords: Predictive analysis · Case-based reasoning · Earth observation · Submarine activity · Space launch.

1 Introduction

In the defense sector, remote sensing, and particularly satellite imagery, is used extensively to detect events on strategic locations. In addition, with the strong increase in the number of commercial satellite images, it has become possible to monitor the activity on specific sites over a long period of time, paving the way for identifying causal patterns on these sites. Such patterns can then be used to predict in advance events that are likely to happen on these locations.

Predictive analysis is a broad field, ranging from forecasting the future values of a series of observations, to detecting events before they actually occur. This topic is very much studied in medicine [2, 10], ecology [4] or finance [9], but very little in remote sensing. Indeed, the majority of prediction methods require to measure data at regular time intervals and to have a large number of past measurements, which is not really possible in remote sensing: depending on the satellite coverage and weather conditions, for a given location, it is very likely that, for a large number of days, no image is available.

In addition, it is often necessary to consider a large number of factors, sometimes on different geographical locations, to understand key patterns and be able to predict certain events. This large increase in the number of features required for prediction makes it particularly difficult for a human operator to forecast events. Hence, the development of automatic algorithms is particularly relevant to predict defense-related events based on satellite imagery.

1.1 Problem setting and contributions

The problem studied here is the detection of specific events from short and irregular series of observations derived from commercial satellite images. The monitored

zones are divided in subareas based on their purposes, for instance administrative areas, road check points, railroads, etc. Times series are then created from the number of objects, such as vehicles or boats, per subarea and per date.

We propose a method for event prediction, using a case-based reasoning approach applied to temporal fragments of satellite image series. This method was designed such that expert insight about the use-case can be directly incorporated, thus facilitating interpretation and pertinence of the results. If the method is meant to be generic, we have focused here on two particular applications: the prediction of the arrival and departure of submarines on a naval base, and the prediction of upcoming rocket launches on launch bases. The proposed case-based reasoning approach has been applied on real data for each use-case, and compares very favorably with a naive approach. From our knowledge, this is the first time that such a prediction method is developed for these applications. We shall stress the fact that very few data were available, making this problem extremely challenging. These results are therefore very promising, and motivate future works on larger datasets to test the method's ability to generalize.

This article is organized as follows: in Section 2 we introduce the two families of methods that are available in the literature for predictive analysis, in Section 3 we describe in details the proposed case-based reasoning approach, while Sections 4 and 5 are dedicated to our experiments on real data, including discussion of the results ; finally, some conclusions are drawn in Section 6.

2 Related work

The problem that we are addressing is located at the junction of several fields, namely event classification, anomaly or rare event detection, and predictive analysis. We can distinguish two main classes of methods. On one hand, we have the methods that see the measured data as a temporal sequence of values which can be modeled in order to predict its values in the future. On the other hand, we have methods that take a step back from the temporal vision of things, cutting the sequence into fragments and treating these fragments independently of their temporal position, which allows to easily reduce the study to a regression or a classification problem.

2.1 Parameterized models

The first strategy used in predictive analysis seeks to directly model the data set in a global way. It assumes that the dataset has a sufficiently regular behavior to be modeled. If the model is manually selected to match the expected behavior, its parameters are automatically estimated to best explain the observations.

Among the many methods using this approach, we can mention the regression methods, where the data are modeled by a parametric function, whose parameters are estimated to fit the observations [12]. These regression methods are used a lot in the medical field where the studied processes (cardiac rhythm, respiratory cycle, etc) are the object of very advanced models, which only need to be parameterized to better fit each patient. Gaussian processes are also widely used for predictive analysis [8]. They provide a probabilistic framework to interpolate and extrapolate temporal sequences while providing additional variance information with the predicted values. However, these methods may not be relevant when feature space is of high dimension and when the number of data samples is low.

More recent approaches use deep learning, in particular the long short-term memory recurrent neural networks [9]. The latter process data sequentially, reusing

some of the network's outputs at the next time step to retain part of the information and thus make it possible to take into account a certain temporal dependence of the observed values. Finally, many models formulate the problem recursively, making each value depend on the previous values. We can thus mention ARIMA (Autoregressive Integrated Moving Average), which assumes a linear dependence involving Gaussian noise terms that is still used today [11], or the Grey model, which is based on the ARIMA model but expressed as a differential equation. However, these methods require regular data and can hardly take into account missing data without serious modifications of the model.

2.2 Sample-based approaches

The second approach consists in separating from the notion of temporality by extracting from the studied time series different temporal segments. Each fragment is treated as a simple point in a high dimensional space and the problem is reduced to a much more studied problem of regression or classification. The objective is then to infer from a given segment if an event is likely to happen, rather than finding the next point of the segment. Various methods are used in this context, for instance support vector machines, which can be associated with the kernel trick for non-linear problems [5]. Other methods are based on fuzzy regression [1], taking into account a certain imprecision on the data. Small neural networks have also been used, taking time series as inputs and outputting the studied value of interest [6].

Case-Based Reasoning (CBR) methods were theorized in the 1990s and have since been used for many applications such as energy demand prediction [7] or financial analysis [3]. This approach is based on using information from a case that is similar to the present one and that has already been solved in the past. This is in contrast to the usual strategy of deducing a set of rules from observations to explain the general behavior of the data. CBR methods require inputs from the user regarding the definition of the manipulated objects:

- Definition of a “case” or “problem”, i.e. the variables allowing to identify a situation, for example for our use cases this definition could involve the location, the date, the number of vehicles of a certain type on a parking lot, etc...
- Defining metrics that allow computing a distance between cases to identify the past cases that are closest to the current one.

The majority of these methods seem to be suitable for the studied problem, provided that the irregularity of the data is taken into account. As explained in the next section, we decided to use the CBR approach.

3 Proposed approach

3.1 Why the CBR?

One specificity of our experiments is that available data are very scarce and irregular: as seen in Sections 4 and 5, between 8 and 33 images are available per use-case, and the time step between two images is not consistent. Since most of the predictive models found in literature use data samples that are regular and abundant, this issue appears to be critical.

One way to overcome data scarcity is to introduce some a priori knowledge, that is, to provide information to the model about the expected behavior or the data. This knowledge can be enforced through optimization constraints during training,

or can be incorporated directly into the problem formulation, for instance by selecting a relevant subset of attributes and thus manually reducing data dimension. Another strategy consists in generating synthetic samples to artificially increase the size of the dataset. However, this approach requires an accurate model to produce realistic data. It shall be noted that even military experts find it difficult to understand the patterns related to the two applications described in Sections 4 and 5. Therefore, in view of the complexity of the use-cases, and the risk to introduce a significant bias in the model, synthetic data generation was not deemed appropriate for our use-cases. Finally, some predictive models are inherently less data greedy, this is the case, for instance, of the Grey model or of CBR, depending on the chosen prediction function.

To carry out this project, we chose to use an approach based on CBR where the prediction function is not parametrized and therefore does not require training. Here, a case is defined by a sequence of dates, each one being associated with a list of scalar attributes, where each attribute is a number of a certain type of objects in a given subarea on the studied location. One advantage of this method is that expert insight can be incorporated into the model by appropriately selecting the object classes and subareas.

3.2 Data interpolation and cross-validation

As mentioned previously, one advantage of the method is that it requires few data to be applied. Nonetheless, to compare two time series we still need them to be regular, i.e. the time interval between two dates must be the same. This condition is not satisfied in our raw data since each date corresponds to a commercial satellite image and this type of imagery is not available with a regular time step. To address this issue we apply a linear interpolation on every time serie. It can be noted that, due to this interpolation, attributes or number of observables can take non integer values for interpolated dates.

In CBR, past cases are used to produce a prediction for the current cases. Hence, if we were to follow the chronological order, early cases would have very few reference cases to make use of and, consequently, their predictions would be less accurate. This is especially problematic here because we have very small datasets. Therefore, to ensure that our experiments are representative enough, we decided to make use of both past and future cases to generate predictions. To perform this cross validation in a relevant way, we set up an overlapping criterion. More precisely, as illustrated in Figure 1, this criterion makes sure that past and future reference cases are far enough from the current case such that the prediction is not directly biased by them.

3.3 Detailed steps and adaptation of CBR

As illustrated in Figure 2, the proposed approach follows four steps:

1. We first select the case to be studied, that is to say a small time series where each date is associated with attributes. The list and definitions of attributes will be detailed for each experiment in Sections 4 and 5. The size of the time series, i.e. the number of days it is made of, is a hyperparameter of the method.
2. For each possible case, past or future, we check if it meets the selection criteria mentioned in Section 3.2, in particular the isolation criterion ensuring that future cases are not too close to the current case. The valid reference cases constitute the case library used for prediction, while the others are discarded.

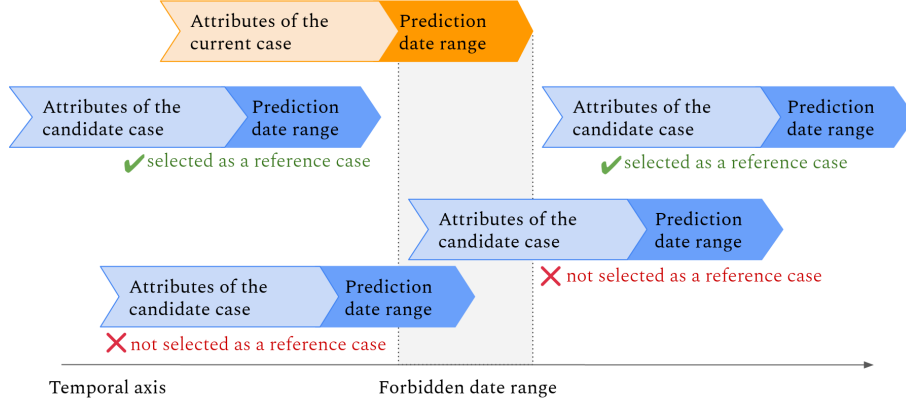


Fig. 1. Illustration of the overlapping criterion for selecting reference cases.

3. Then, the Euclidean distance between each reference case and the case under study is computed. The K nearest neighbors are selected as being the most resembling cases, K being set by the user.
4. Finally, the prediction y for the case under study is obtained by computing the weighted average of the ground-truths of the K nearest reference cases:

$$y = \frac{\sum_{i=1}^K p_i \exp\left(-\frac{d_i^2}{2\sigma^2}\right)}{\sum_{i=1}^K \exp\left(-\frac{d_i^2}{2\sigma^2}\right)}, \quad (1)$$

where p_i is the ground-truth of the i th reference case and σ is manually set to a percentage of the data standard deviation (20% in practice). Hence, the closer a reference case is to the one under study, the more weight it will have in the prediction. As seen in Eq. (1), weights are obtained thanks to a Gaussian kernel, so that the influence of cases that are farther away decreases faster. On the choice of σ : the larger it is, the more y will tend to a simple average, and on the contrary, if σ is small then only the d_i corresponding to the closest cases will be taken into account.

In the next sections, the proposed CBR approach is applied on two different use-cases.

4 Predicting submarines arrivals and departures

The first use-case addresses the prediction of imminent arrivals and departures of submarines on a naval base, which is referred to as naval-base-1.

Data – For this experiment, we selected commercial satellite images based on two constraints: 1/ these images had to contain very few clouds, such that submarines could be identified, and 2/ they needed to be close to each other on the temporal axis, since the proposed predictive method makes use of time series. We acquired all available images of naval-base-1 satisfying these criteria, leading to a total of 33 images, with 3 images in June 2018, 3 in September 2019 and 27 in 2020.

Feature extraction – The area of interest in naval-base-1 is divided into two zones based on the analysis of experts at Preligens. For each zone, we manually

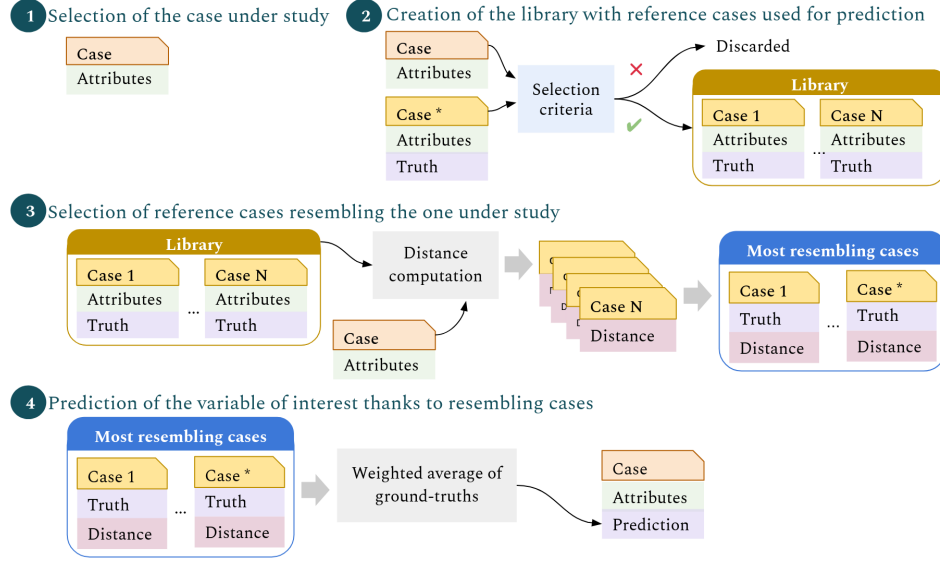


Fig. 2. Steps involved in the proposed CBR method.

determined the number of vessels belonging to the following 9 classes for every date: relevant submarine classes (Delta III, Oscar II, Borei and Akula), warships, support ships (including tugboats), barge-mounted cranes, speedboats and civilian boats. In addition, zone 2 also exhibited a parking lot, so we counted the number of vehicles on this area for each image.

Formulation of the prediction task – Given the nature of the studied naval base, the following two variables were defined as prediction targets:

- the arrival of at least one submarine of all classes in the next 4 days,
- the departure of at least one submarine of all classes in the next 5 days.

It shall be noted that we are not interested in predicting the number of submarines, since this variable could stay constant even if there have been multiple departures and arrivals. The separation between the two tasks above enriches the study, and is made possible thanks to the precise identification of each submarine’s class. In addition, we are not trying to estimate the number of arrivals and departures, but rather to predict if at least one such event is likely to happen. This choice is motivated by the fact that only a small amount of data is available, which makes this study challenging. Also, we do not have one image for each day in the studied period, so the ground-truths we used for the predicted events are approximations. Indeed, in the days following the prediction date, there could be an arrival or departure of submarine on a date for which no image is available.

Hyperparameters – The proposed CBR approach includes several hyperparameters that are chosen manually. Several configurations were tried and we kept the one leading to the best predictions. Regarding the prediction support, which is the number of previous days used to estimate the likelihood of the studied events, we used time series of 9 days preceding the prediction date. For both arrivals and departures we used a total of 5 nearest neighbours to make the prediction. Finally, each attribute is defined by a pair (class, area), where classes could also be merged categories. From all possible attributes we kept only the most relevant ones: the

number of elements for each one of the four submarine classes in zone 1, the number of warships in zone 1 and the number of boats (all categories are mixed) in zone 2. Regarding the arrival of submarines, we included an additional attribute, which is the number of vehicles in zone 2.

Results and discussion – To complement the study, we compared the proposed CBR method with a random approach, where the neighbors that are selected in the library to infer the prediction are chosen randomly before applying Eq. (1). For this comparison, we take the average of 10 random draws. Results for both the proposed approach and the random method are presented in Table 1. The proposed CBR model correctly predicts all actual arrivals and departures in zone 1, and leads to only two false alarms: one arrival and one departure that are expected by the model but not visible in the images.

Table 1. Prediction of submarines activity on naval-base-1. True positive: an event is correctly predicted. True negative: the absence of event is correctly predicted.

	Submarine arrivals (4 days)		Submarine departures (5 days)	
	True positives	True negatives	True positives	True negatives
Nearest neighbors	3/3	7/8	6/6	7/8
Random	2.5/3	6.5/8	3.7/6	4.7/8

In addition, the CBR approach compares well with the random draws, which suggests that the model positively identifies "patterns" in the data that help producing correct predictions. To check whether these patterns are causal or just lucky correlations, we carry out an ablation study. In other words, we gradually remove attributes and study how the prediction is affected. For the submarines arrivals, this process suggests that most of the information is encased in the number of vehicles in zone 2. From an operational point of view, zones 1 and 2 do not serve the same purpose, thus the pattern used by the model to predict arrivals is likely to be a lucky correlation in this case. For departures however, the information seems to be diluted in the different attributes that we selected. Therefore, the associated patterns might be more relevant and would benefit from a more in-depth study if more data is available.

It is worth noticing that other configurations were studied, some including for instance the number of barge-mounted cranes in the list of attributes. However, these additional attributes, despite being relevant from an operational point of view, did not raise the performance. This might be explained by their small representation in the data and the restricted number of images that were available.

5 Predicting rocket launches

This second use-case aims at predicting imminent rocket launches on two different space launch bases, namely space-base-1 and space-base-2.

Data – We retrieved online the official dates of past launches on the two locations, and gathered all available sequences of commercial satellite images close to these dates. We were able to acquire 8 images for space-base-1 and 14 images for space-base-2, that were associated to 3 launch dates for both locations.

Feature extraction – The definition of relevant areas in the two launch bases was provided by a group of experts at Preligens. Then, using Preligens’ vehicle detectors, we were able to automatically get the number of vehicles in each sub-area for every image. In addition, for space-base-1 we manually annotated some elements that might be of interest from an operational point of view.

Formulation of the prediction task – The goal of the CBR model for this application is to predict an imminent rocket launch on the current date or in the next 4 or 5 days on space-base-1 and space-base-2, respectively. Here, we do not try to predict the launch pads on which the event will happen. There are two reasons for this: first, in view of the small amount of data we would have a very low number of events per launch pad, and second, this information is not always publicly available. In addition, we are not trying to predict a precise launch date since the database would not be sufficient for a problem of that complexity.

Hyperparameters – The support size is 1 for this application: we use the image of only one date to infer whether a launch is likely to happen in the next days. Increasing the support size would lead to very few valid cases in view of our overlapping criterion illustrated in the Figure 1. For every prediction we select 3 nearest neighbors in the case library. We select the attributes that are most relevant to the application and that lead to the best results:

- for space-base-1 we use 6 attributes, including the number of vehicles in 5 different areas (administration, preparation, launch pads, maintenance, road check-point)
- for space-base-2 we use the number of vehicles in 3 subareas (administration, preparation and launch areas).

For space-base-1, the number of vehicles per subarea is in average much higher than the last attribute. Therefore, in order to avoid an imbalance in the computation of the Euclidean distance, we artificially divide the number of vehicles on space-base-1 by 10. This can be viewed as a type of normalization.

It shall be noted that the output of the model is a number between 0 and 1. For the prediction of submarines arrivals and departures, the output was numerically sufficiently close to either 0 or 1 so that we did not need to tune a threshold to decide whether the final prediction was 0 or 1. Here, it is different, predictions can take intermediary values. Hence, we chose to take a threshold equal to 0.1 to avoid missing launches: if the launch score is above 0.1, then we set the prediction to 1.

Results and discussion – Results are presented in Table 2, similarly to the previous use-case, we include a comparison with a model using random selection of neighbors. The proposed approach leads to one false alarm on space-base-1, while all 14 predictions are correct on space-based-2. The CBR model compares favorably with the random approach: for space-base-1 the latter leads to more than 2 false alarms in average, and for space-base-2 it misses more than half the launches. This suggests that the model identifies relevant patterns.

Regarding the first location, the ablation study clearly demonstrated that mainly one attribute, which is relevant from an operational point of view, is the key pattern. For space-based-2, the ablation studies shows that the attribute that provides less information is the number of vehicles in the administrative areas: when considering only this attribute, the model makes 3 mistakes compared to none when the other dimensions are used individually. This could be explained by a low number of vehicles on average in administrative areas (usually below 5), when other areas feature larger variations, from one to several dozens of vehicles

Table 2. Results for the prediction of imminent rocket launches. True positive: an event is correctly predicted. True negative: the absence of event is correctly predicted.

	space-base-1 (4 days)		space-base-2 (5 days)	
	True positives	True negatives	True positives	True negatives
Nearest neighbors	5/5	2/3	6/6	8/8
Random	4.0/5	0.7/3	2.9/6	7.0/8

**Fig. 3.** Number of vehicles before and after a launch on space-base-2. Close-ups of preparation (first row) and launch (second row) subareas. Indicated numbers of vehicles are taken on the full subareas so not all vehicles are visible.

depending of the date. Fig. 3 illustrates this phenomenon: we can see that the number of vehicles increases dramatically ahead of launches.

Here, the patterns that seem to be exploited by the model can be inferred by a human as well: we are dealing with few images and the evolution of the observables of interest the days prior to a launch is quite obvious. Hence, we do not claim that this model has discovered unknown patterns, but rather that the behavior of the proposed method is reassuring and in accordance with operational consideration, showing its potential for other, more complex, applications.

6 Conclusion

In addition to being applicable to very irregular time series, the proposed approach has the benefit of easing the interpretation of the prediction, as the selection of the nearest neighbors and their distance from the study case allows a better understanding of the patterns present in the data and related to the variable of interest to be predicted. If the results of the method on the two use cases presented are very promising, it is important to note that the small amount of data used does not allow to fully validate the method. To do so, it would be necessary to perform additional tests on larger data sets.

Moreover, to obtain a prediction value from the neighbors, we use here a simple weighted average. If the large size and the small number of data prevent us from using advanced methods (such as deep neural networks), other methods (such as SVM) could be used in future works to bring more expressivity to the model. Finally, it could be interesting to study the use of other distance measures less sensitive to missing data (using for example probability distributions) or which could take into account temporal deformations of the studied patterns (in the line of the dynamic time warping).

Acknowledgment

This work is supported by the French "Direction Générale de l'Armement" through the research project RAPID S-Cube in collaboration with Preligens.

References

1. Azadeh, A., Saberi, M., Seraj, O.: An integrated fuzzy regression algorithm for energy consumption estimation with non-stationary data: a case study of iran. *Energy* **35**(6), 2351–2366 (2010)
2. Beyene, C., Kamat, P.: Survey on prediction and analysis the occurrence of heart disease using data mining techniques. *International Journal of Pure and Applied Mathematics* **118**(8), 165–174 (2018)
3. Corchado, J.M., Lees, B.: A hybrid case-based model for forecasting. *Applied Artificial Intelligence* **15**(2), 105–127 (2001)
4. Dietze, M.: *Ecological forecasting*. Princeton University Press (2017)
5. Fathima, A., Manimegalai, D.: Predictive analysis for the arbovirus-dengue using svm classification. *International Journal of Engineering and Technology* **2**(3) (2012)
6. Pao, H.T.: Comparing linear and nonlinear forecasts for taiwan's electricity consumption. *Energy* **31**(12), 2129–2141 (2006)
7. Platon, R., Dehkordi, V.R., Martel, J.: Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. *Energy and Buildings* **92**, 10–18 (2015)
8. Richardson, R.R., Osborne, M.A., Howey, D.A.: Gaussian process regression for forecasting battery state of health. *Journal of Power Sources* **357**, 209–219 (2017)
9. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing* **90**, 106181 (2020)
10. Shinde, G.R., Kalamkar, A.B., Mahalle, P.N., Dey, N., Chaki, J., Hassanien, A.E.: Forecasting models for coronavirus disease (covid-19): a survey of the state-of-the-art. *SN Computer Science* **1**(4), 1–15 (2020)
11. Siami-Namini, S., Namin, A.S.: Forecasting economics and financial time series: Arima vs. lstm. *arXiv preprint arXiv:1803.06386* (2018)
12. Zubakin, V.A., Kosorukov, O.A., Moiseev, N.A.: Improvement of regression forecasting models. *Modern applied science* **9**(6), 344 (2015)

End-to-End Pipeline for Visually Rich Documents Comprehension using Deep Learning applied to Nuclear Industry^{*}

Aleksei Iancheruk¹, Ahmed Allali², Julien Rodriguez³, Tejas Bhor⁴, Ricardo Garcia⁵, Jean-Eudes Guilhot-Gaudeffroy⁶, and Robert Plana⁷

Assystem, France^{1,2,3,4,5,6,7}

{aiancheruk,aallali,jrodriguez,tbhor,jrduartegarcia,jeguilhot,rplana}@assystem.com

Abstract. During the construction and operation of a Nuclear Power Plant (NPP), a large mass of documentation is produced. Key events in the life of a NPP like installation of a new equipment or upgrade of an existing one are resumed by various reports. Even though in most of the cases these reports follow predefined templates, their reading and verification is time-consuming as they are often paper-based, low quality and contain handwritten comments made by engineers during their inspections. Existing solutions are not performant enough to process such challenging data, use too much computational resources for training and can't be used in nuclear field which is imposing high expectations when it comes to data privacy. In this study, we present a generic pipeline for reports and other visually rich unstructured documents understanding using established Deep Learning state-of-the-art approaches. A key contribution of our research is that such solution can be adapted to new data in short time, with a minimal annotation effort and limited computational resources. We evaluate our method on key-value information extraction task. Experiments show that the proposed method demonstrates high performance across various document templates.

Keywords: Documents Intelligence · Semantic Segmentation · Nuclear · Key-value extraction · Optical Character Recognition.

1 Introduction

Many systems of NPP are designed to operate for its entire service life. This means that the engineers have to face huge amounts of legacy documentation that don't yet come in digital form. Most of this documentation are inspection reports and other visually rich documents similar to forms. These documents describe the works that have been done, and onsite teams are often required to manually verify them to ensure the conformity of the installed equipment to project requirements. Portable Document Format (PDF) is one of the most popular formats that is used for such documents. While this format is convenient for human understanding, it is not suitable for automation processes. To

^{*} Supported by Assystem Engineering & Operation Services, France.

monitor systems and measure their aging status, engineers need to have access to information contained in these documents. In case of anomalies or in need of additional calculations, they have to consult hundreds of pages to track down the possible issue. Documents in PDF format don't allow easy access to it. To facilitate work, engineers are often obliged to manually transform the documents they are working with to digital form suitable for export to tabular formats, database queries and calculations.

Multiple Deep Learning methods were proposed to analyze unstructured documents using only their visual structure [1]. They are using object detection or instance segmentation to extract regions of interest on the page. When extracted, these regions can be analyzed with Optical Character Recognition (OCR) to get their text if the document is not digital-born. The main drawback of these methods is that they require large datasets of high variety training data (different templates, etc.) This data should be annotated manually, and it is often engineers who need to accomplish it, which makes it a slow and expensive process.

In this work, we propose a method for end-to-end comprehension of visually rich documents using limited annotation and computational resources, which are often seen in the context of the nuclear industry. We accomplish it by using (1) as a first step - template classification allowing to find pages with relevant information, (2) instance segmentation model allowing to extract key-value information regions and (3) custom OCR models for printed and handwritten text recognition.

Our experiments illustrate that methods based on convolutional networks alone without taking into consideration textual features are very effective for extraction of regions of interest from visually rich documents, such as equipment installation reports. Our method also respects data privacy, as we don't make calls to any third-party services.

2 Pipeline Architecture

In this section, we briefly present our solution architecture. Fig. 1 illustrates its general overview. Each pipeline step is preceded by an annotation step. To reduce annotation effort, we propose to use a semi-automatic approach. The workflow is the following:

- i annotation of seed data - a small subset of expected training samples (10% to 20% of total expected number of training samples);
- ii training of the model on annotated seed data with a decreased number of iterations;
- iii prediction on the next subset of data using the model trained on seed data;
- iv manual correction of annotations produced by the model;
- v retraining of the model on all annotated samples using complete training schedule.

The process is repeated till all the training data is annotated or a desirable level of performance is achieved on the test dataset. Such approach accelerates the

slow annotation process and reduces the number of annotated images needed for training.

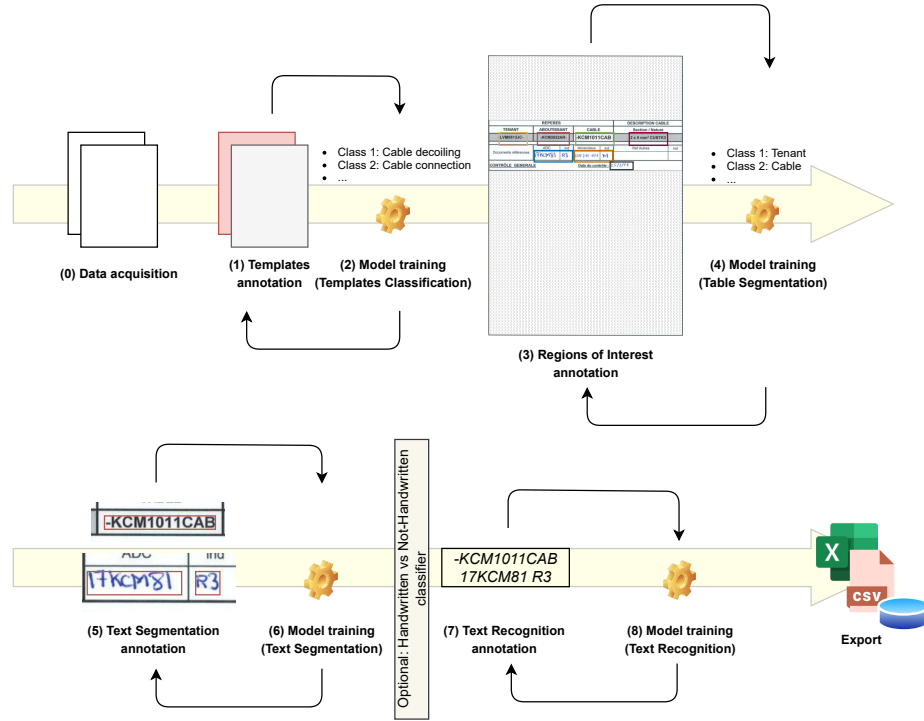


Fig. 1. An overview of the pipeline architecture.

2.1 Data Preparation

In this study, we propose to focus on only the PDF format files, as they are the most commonly used in the nuclear industry. The size of each PDF file we process for our experiments can be ranged from 20 to 200 pages. The total volume of documents is 1700 PDF files. All the documents are image based and more than 15% of them contain handwritten comments.

As a first step to our pipeline, we render each page of PDF file as an image having sufficient resolution (more than 1500 by 2000 pixels). No textual information is taken from PDF, and no other preprocessing is done on the images.

2.2 Templates Classification

The template classification module is an image classification model based on ResNet34 [2] backbone pretrained on ImageNet [3]. The goal of this module is

to classify each page of the document in a number of predefined classes. This allows to filter out pages which don't contain relevant information and to apply adapted logic to each of the detected page classes if needed. We replace the final FC (fully connected) layer of pretrained ResNet34 with the FC layer that outputs the number of needed classes.

Equipment installation reports we are working with in this study are visually rich documents and therefore contain different layouts that can be distinguished by image-based approaches. An important note is that such approach can't be applied to, for example, free-form documents like specifications and letters, as they will normally need content-based classification using textual features.

2.3 Key-value Extraction

The key-value extraction module is an instance segmentation model based on Mask R-CNN [4] with ResNeXt101 followed by FPN [5, 6] as a backbone and FC heads for mask and box prediction pretrained on ImageNet and COCO [7] datasets. Compared to methods which are segmenting whole tables to extract information from forms and similar documents [8], we propose to extract only relevant regions of interest which will be useful to transform the document to a digital form. As a result of this extraction, each pixel of the document page image will be assigned a label.

As mentioned before, relevant information in reports is often represented as key-value pairs. In our study, however, we ignore annotation of the key region as we predict directly the value with its class.

2.4 Text Segmentation

Existing OCR systems can't handle both printed and handwritten text detection, as well as detection of small text regions (1-2 characters). Therefore, for this step, we propose to use a custom model trained for the task of detection of both printed and handwritten characters.

The model that we use for this step is the same architecture as used for the key-value extraction module. It will take value regions from the previous step as inputs and predict masks of each character for each object region.

As an optional step, we also perform classification of each character in handwritten and printed classes. This will allow increasing recognition accuracy as two separate models (for printed and handwritten characters) can be further applied depending on the character's class. The classifier we use is a multi-layer convolutional network with a FC layer on top.

2.5 Text Recognition

The text recognition is a module for transforming extracted text regions to text. We are using four-stage architecture with thin-plate spline (TPS) transformation, a ResNet32 for feature extraction, Bidirectional LSTM (BiLSTM) for

sequence modelling stage and attention-based sequence prediction at prediction stage [9]. In this study, we present results of using this architecture applied to handwritten and printed characters recognition.

3 Experiments

In this section, we present the experimental setup and results for our pipeline. We don't include experiments results for all the modules but only for the most critical ones. To simulate limited hardware resources, all experiments were done on a single Tesla M60 GPU with 8Gb of memory.

3.1 Experiments for Key-value Extraction

Data and Setup We use our annotated dataset of cables equipment reports. The final version of the dataset contains 1088 images for training and 272 images for testing. The dataset was obtained using our iterative annotation process, starting with 10% of the total training images as a seed data. The dataset comes in COCO annotation format. Regions of interest are classified in 11 classes {aboutissant, cable, date, fad, nomenclature, nomenclature-ind, plan-aboutissant, plat-tenant, reserve, section and tenant}. The overall metric is mean average precision (MAP) @ intersection over union (IOU) [0.50:0.95] of bounding boxes.

We trained a Mask R-CNN model on our dataset using the Detectron2 [10] implementation from Facebook Research. The PDF pages were converted to images using PyMuPDF package. We fine-tuned ResNeXt-101-32x8d-FPN pre-trained on ImageNet and COCO dataset (3x schedule) on our custom dataset for 3k iterations with a learning rate of 0.0009.

Results The performance of the Mask R-CNN model on our testing set is depicted in Table 1. The fine-tuned model can predict accurate bounding boxes and masks for most relevant classes of regions of interests (MAP > 0.65) using only visual features of the documents. Higher results for some classes (MAP > 0.70) can be explained by a higher number of training data for them.

Table 1. MAP @ IOU [0.50:0.95] of the Mask R-CNN for Key-value Extraction model.

Category	M-RCNN	Category	M-RCNN	Category	M-RCNN
aboutissant	67.26	cable	70.49	date	62.49
fad	41.21	nomenclature	65.72	nomenclature-ind	50.95
plan-aboutissant	57.08	plan-tenant	59.48	reserve	71.07
section	67.27	tenant	69.67		

3.2 Experiments for Text Segmentation

Data and Setup Like for key-value extraction, we use a dataset annotated by our means. It contains 1944 images for training and 216 images for testing. Annotations come in COCO format. The dataset contains both printed and handwritten text regions. The metric is MAP @ IOU [0.50:0.95].

The same model as for key-value extraction was used for fine-tuning. It was fine-tuned for 3k iterations with a learning rate of 0.001.

Results The performance for this model is depicted in Table 2. Text segmentation achieves the best performance of MAP 65.25. A less strict metric of AP @ IOU 0.75 gives us an AP of 79.43 which we believe a good result given the fact that we train to detect both printed and handwritten text.

Table 2. MAP @ IOU [0.50:0.95], AP @ IOU 0.50 and AP @ IOU 0.75 of the Mask R-CNN Text Segmentation model.

Metric	M-RCNN
MAP @ IOU [0.50:0.95]	65.25
AP @ IOU 0.50	97.31
AP @ IOU 0.75	79.43

3.3 Experiments for Text Recognition

Data and Setup For this module, three types of datasets were produced: (1) dataset with handwritten digits only, (2) dataset with handwritten letters only and (3) dataset with printed text. The handwritten digits dataset contains 2326 training images and 1371 test images, while the handwritten letters dataset contains 2943 training images and 1857 test images. As for the printed text dataset, it contains 6811 training images and 1703 test images. The metrics for this module’s experiments are accuracy and edit distance.

Three models were tested: (1) model that predicts handwritten digits only (12 characters), (2) model that predicts handwritten letters (53 characters) and (3) model that is case-sensitive and predicts printed digits, letters and special characters (94 characters). All models are based on TRBA architecture, which we have described in previous sections. The model for letters was trained for 6k iterations, while the model for digits for only 1k iterations and printed text model had 10k training iterations. All models had a learning rate of 0.1.

Results The performance for text recognition models is presented in Table 3. As we expected, for the case of handwritten text, the model for digits recognition outperforms the one for letters recognition. The letters dataset is more difficult for the model and more iterations are needed to achieve similar level

of digits model’s performance. Both models, however, achieve a very high level of performance overall, with an accuracy higher than 90.0 and an edit distance higher than 0.90. As for printed text, we can see that currently existing open-source text recognition tools (Tesseract 4.1.1, EasyOCR 1.4) have insufficient robustness which is noticed mainly with degraded images and images with colored background as illustrated in Table 4. This shows that these tools can’t be used, at big scale, for degraded text recognition.


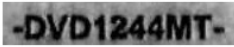
Table 3. Accuracy and Edit Distance of handwritten and printed characters recognition models (letters and digits) compared to existing OCR tools.

Metric	TRBA (ours) Handwritten letters	TRBA (ours) Handwritten digits	TRBA (ours) Printed all	EasyOCR Printed all	Tesseract Printed all
Accuracy	91.10	98.65	97.94	26.78	83.56
Edit Distance	0.92	0.99	0.99	0.88	0.96

3.4 Pipeline Performance

In order to assess the global performance of our solution, we calculated the accuracy per page on all classes and, depending on text type frequency (printed or handwritten), we provide a mean global accuracy per page. Our solution demonstrates an accuracy of 86.21 which in our opinion is reliable enough to handle and process complicated and diverse document templates with minimal manual verification.

Table 4. Recognition models result samples.

Value region	Ground truth	Prediction (Ours)	Tesseract	EasyOCR
	-GEV315201	-GEV315201	-	-
	-DVD1244MT-	-DVD1244MT-	-	DVDIZAaMI:

4 Conclusion and Future Work

In this study, we presented an end-to-end approach for visually rich document understanding applied to nuclear equipment reports using state-of-the-art com-

puter vision methods. The availability of the annotated data remains a bottleneck for domains like nuclear. Our approach allows faster iterations of data annotation and doesn't require large datasets to achieve high level of performance. Using orders of magnitude less data, engineers will be able to digitalize large amounts of low-quality legacy documentation. The main limitation of such approach remains dependence on the visual structure of a document, thus it will not be possible to generalize on the documents which may be free form or mixed type. To go further, for future research we may investigate multi-modal approaches like the one used by [11] taking into consideration both visual and textual features to process more types of documents.

References

1. Wang, J., Liu, C., Jin, L., Tang, G., Zhang, J., Zhang, S., Wang, Q., Wu, Y., Cai, M.: Towards Robust Visual Information Extraction in Real World: New Dataset and Novel Solution. arXiv:2102.06732 [cs]. (2021).
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs]. (2015).
3. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575 [cs]. (2015).
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. arXiv:1703.06870 [cs]. (2018).
5. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated Residual Transformations for Deep Neural Networks. arXiv:1611.05431 [cs]. (2017).
6. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. arXiv:1612.03144 [cs]. (2017).
7. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs]. (2015).
8. Paliwal, S., D, V., Rahul, R., Sharma, M., Vig, L.: TableNet: Deep Learning model for end-to-end Table detection and Tabular data extraction from Scanned Document Images. arXiv:2001.01469 [cs, eess]. (2020).
9. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. arXiv:1904.01906 [cs]. (2019).
10. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2, <https://github.com/facebookresearch/detectron2>, (2019).
11. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: Pre-training of Text and Layout for Document Image Understanding. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1192–1200 (2020). <https://doi.org/10.1145/3394486.3403172>.