Actes de la conférence CAID 2025

(Conference on Artificial Intelligence for Defense)

Organisée par

Sim2Real SAR Image Restoration: Metadata-Driven Models for Joint Despeckling and Sidelobes Reduction

Antoine De Paepe[1], Pascal Nguyen[2], Michael Mabelle[2], Cédric Saleun[1], Antoine Jouadé[1], and Jean-Christophe Louvigne[1]

[1]Direction Générale de l'Armement Maîtrise de l'Information, Bruz, France.
[2]Agence Ministérielle pour l'Intelligence Artificielle de Défense, Bruz, France.

***Abstract*—Synthetic aperture radar (SAR) provides valuable information about the Earth's surface under all weather and illumination conditions. However, the inherent phenomenon of speckle and the presence of sidelobes around bright targets pose challenges for accurate interpretation of SAR imagery. Most existing SAR image restoration methods address despeckling and sidelobes reduction as separate tasks. In this paper, we propose a unified framework that jointly performs both tasks using neural networks (NNs) trained on a realistic SAR simulated dataset generated with MOCEM. Inference can then be performed on real SAR images, demonstrating effective simulation to real (Sim2Real) transferability. Additionally, we incorporate acquisition metadata as auxiliary input to the NNs, demonstrating improved restoration performance.**

***Index Terms*—SAR, Image Restoration, Despeckling, Sidelobes Reduction, Metadata Injection, Sim2Real**

Notation and Terminology

Throughout this paper, the term SAR image refers broadly to SAR data, including both single look complex (SLC) images and their amplitude representations. When necessary for clarity, we explicitly distinguish between the two. For Section II, all vectors are represented as column vectors. The symbol '$\top$' denotes the matrix transpose operation. For a given column vector $\boldsymbol{s} = [s_1, s_2, \cdots, s_n]^\top \in \mathcal{S}^n$, with $\mathcal{S}$ being either $\mathbb{R}$ or $\mathbb{C}$, s_k denotes the kth entry of $\boldsymbol{s}$. For the specific case where $\mathcal{S} \triangleq \mathbb{C}$, we write $\boldsymbol{s} = \boldsymbol{s}^{\text{re}} + i\boldsymbol{s}^{\text{im}}$, with $\boldsymbol{s}^{\text{re}} \in \mathbb{R}^n$ denoting the real part and $\boldsymbol{s}^{\text{im}} \in \mathbb{R}^n$ denoting the imaginary part. We define $\boldsymbol{D_s} \triangleq \text{diag}(\boldsymbol{s}) \in \mathcal{S}^{n\times n}$ as the diagonal matrix whose diagonal entries are given by the components of the vector $\boldsymbol{s}$.

I. Introduction

SAR imaging has emerged as a crucial remote sensing technology, extensively applied across a range of domains, including environmental monitoring, disaster management, and military surveillance. Its capability to acquire high-resolution imagery under various conditions renders SAR a valuable tool for observing and analyzing the Earth's surface.

Despite its many advantages, SAR imagery is often compromised by inherent noise—particularly speckle—as well as sidelobes surrounding bright targets, which can obscure critical details. To improve the quality of SAR image interpretation, it is essential to address the challenges posed by both speckle noise and the presence of sidelobes.

A variety of sidelobes reduction techniques have been developed, each targeting different aspects of SAR signal processing. The most widely used methods are windowing functions [1], such as Hamming, Hanning, and Kaiser windows, which reduce sidelobes amplitude by smoothing the radar signal during processing, albeit at the cost of some resolution loss. More sophisticated approaches aim to minimize sidelobes levels without significantly compromising resolution or signal strength, such as spatial variant apodization (SVA) [2]. However, SVA presents several drawbacks: it distorts statistics in speckle-dominated regions, spreads point-like targets across multiple pixels, and introduces a negative bias, making homogeneous areas appear less reflective [3]. Recent advancements in deep learning have introduced convolutional neural network (CNN)-based [4] and bidirectional recurrent neural network (RNN)-based [5] models to mitigate these issues.

Parallel to sidelobes reduction techniques, SAR image despeckling has been extensively studied and is now commonly addressed through two main categories of methods: variational approaches and learning-based techniques.

Variational methods typically formulate the problem as the minimization of a cost function balancing a data fidelity term—which accounts for the statistical properties of speckle under the Goodman speckle model [6]—and a regularization term, which incorporates prior knowledge of the underlying image [7, 8].

Deep learning-based methods are usually trained on natural image pairs augmented with synthetic speckle noise to effectively learn the despeckling task. SAR-CNN [9] and ID-CNN [10] were specifically designed to handle the multiplicative nature of speckle noise, inspired by the DnCNN architecture [11]. Additionally, UNet architectures [12], initially developed for medical image segmentation, have proven highly effective in natural image restoration tasks [13, 14, 15], and their adaptability has been demonstrated in SAR despeckling applications [16, 17]. These networks excel at capturing multi-scale features, which is crucial for restoring fine details in noisy SAR images. Generative adversarial networks (GANs) have

also emerged as a powerful tool for SAR image restoration [18, 19]. A more recent development in SAR image restoration is the application of conditional diffusion processes [20, 21] for despeckling SAR images. Finally, self-supervised learning approaches [22, 23], which eliminate the reliance on ground truth (GT) data, are emerging as effective solutions for SAR image restoration.

Despite the promise of deep learning methods, most are primarily designed to reduce spatially uncorrelated speckle due to their training strategies. However, in real SAR images, speckle is often spatially correlated through the SAR transfer function. This domain gap can lead to suboptimal restoration in practical applications. Few methods have been developed to address this challenge, such as MuLoG-DRUNet [24], a plug-and-play (PnP) method that stands out as the first robust technique to reduce spatially correlated speckle.

In this paper, we propose a novel approach for joint despeckling and sidelobes reduction for SAR image restoration using deep learning-based methods. We use MOCEM [25] to create a dataset of SAR images with access to GT, specifically designed to replicate the speckle characteristics and effects of the SAR transfer function observed in real-world scenarios. This dataset is used for a supervised learning task, and its consistency with real SAR acquisitions enables effective Sim2Real SAR image restoration. Furthermore, we propose incorporating SAR acquisition metadata to enhance the restoration process. By integrating metadata into the deep learning framework, we provide contextual information that allows the NNs to tailor its restoration strategy according to specific acquisition conditions.

II. Methodology

A. Problem Formulation

The SAR acquisition process involves a sensor, typically mounted on an aircraft or satellite, which acts as an active system that emits electromagnetic waves toward the Earth's surface and captures the reflected signals as a set of complex-valued measurements. These measurements are coherently combined to synthesize a long virtual aperture, forming the SLC image $\boldsymbol{y} \in \mathcal{Y} \triangleq \mathbb{C}^n$ of the scene reflectivity $\boldsymbol{x} \in \mathcal{X} \triangleq \mathbb{R}^n_+$ being imaged. For interpretability and practical processing, SAR images are also expressed in terms of the amplitude of the SLC, defined as $\tilde{\boldsymbol{y}} \triangleq |\boldsymbol{y}|$, with $\tilde{\boldsymbol{y}} \in \tilde{\mathcal{Y}} \triangleq \mathbb{R}^n_+$. Owing to the impulse response of the SAR system and the coherent nature of radar signals, SAR images are often degraded by both sidelobes and spatially correlated speckle. We denote by $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ the linear spatial-domain operator associated with the SAR transfer function. Under the Goodman model, it is commonly assumed that for a restoration task, the measurement $\boldsymbol{y}$, in the absence of electronic noise, follows a circular Gaussian distribution [22] with independent entries given by

$$(\boldsymbol{y}|\boldsymbol{H},\boldsymbol{x}) \sim \mathcal{CN}(\boldsymbol{0}_{\mathcal{X}}, \boldsymbol{H}\boldsymbol{D}_{\boldsymbol{x}}\boldsymbol{H}^{\top}), \tag{1}$$

where $\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a complex circular Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$. This model can be reparametrized to exhibit the multiplicative nature of speckle as

$$\boldsymbol{y} = \boldsymbol{H}(\mathcal{C}(\boldsymbol{x}^{\frac{1}{2}}) \odot \boldsymbol{n}), \quad \boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}_{\mathcal{X}}, \boldsymbol{I}_{\mathcal{X}}), \tag{2}$$

where $\mathcal{C} : \mathbb{R}^n \to \mathbb{C}^n$ is the complexification operator such that $\mathcal{C}(\boldsymbol{x}) = \boldsymbol{x} + i\boldsymbol{x}$, and $\boldsymbol{n}$ represents the multiplicative speckle noise in its complex form.

When the transfer function $\boldsymbol{H}$ is known, image restoration can be achieved by finding the maximum a posteriori (MAP) estimate of $\boldsymbol{x}$, i.e.,

$$\max_{\boldsymbol{x}\in\mathcal{X}} p(\tilde{\boldsymbol{y}}|\boldsymbol{H},\boldsymbol{x}) \cdot p(\boldsymbol{x}), \tag{3}$$

where the conditional probability density function (PDF) $p(\tilde{\boldsymbol{y}}|\boldsymbol{H},\boldsymbol{x})$ is given by (1), and $p(\boldsymbol{x})$ is a prior PDF on $\boldsymbol{x}$, which is generally unknown and replaced (in its post-log form) by a regularizer promoting piecewise smoothness. However, solving the inverse problem in (3) relies on knowledge of $\boldsymbol{H}$, which may be unavailable in many real-world applications—particularly when the processing pipeline or acquisition geometry is proprietary or undisclosed. In such cases, traditional model-based restoration approaches become inapplicable or unreliable due to the absence of an accurate system model, especially in the joint task of despeckling and sidelobes reduction.

To address this limitation, we consider an alternative strategy that leverages a large collection of known SAR transfer functions to train NN $f_{\boldsymbol{\theta}} : \tilde{\mathcal{Y}} \to \mathcal{X}$, parameterized by $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, which can generalize to unseen or implicitly characterized systems. The training process is formulated as the minimization problem

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbb{E}_{(\boldsymbol{x},\tilde{\boldsymbol{y}})\sim p(\boldsymbol{x},\tilde{\boldsymbol{y}})}[\ell(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{y}}),\boldsymbol{x})], \tag{4}$$

where $p(\boldsymbol{x},\tilde{\boldsymbol{y}}) = \int p(\tilde{\boldsymbol{y}}|\boldsymbol{H},\boldsymbol{x})p(\boldsymbol{H})p(\boldsymbol{x})d\boldsymbol{H}$ is the joint PDF of $\boldsymbol{x}$ and $\tilde{\boldsymbol{y}}$, marginalized over $p(\boldsymbol{H})$, the PDF of possible SAR transfer functions, and $\ell : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a loss function. In this way, the model implicitly learns to invert the composite effect of the transfer function $\boldsymbol{H}$ and the speckle noise. This approach enables model-free SAR image restoration, particularly suited to operational contexts.

B. Sim2Real: Learning with Simulations for Real-World Restoration

In real-world scenarios, obtaining a large dataset of paired samples $(\boldsymbol{x},\tilde{\boldsymbol{y}}) \sim p(\boldsymbol{x},\tilde{\boldsymbol{y}})$ for supervised learning is nearly impossible. It therefore becomes necessary to overcome this limitation through simulation by generating synthetic training datasets. We propose using simulated SAR images obtained with the MOCEM [25] software. This simulation tool for modeling SAR acquisition enables complete replication of the SAR imaging chain, producing high-fidelity SAR images from computer aided design (CAD) models while incorporating fundamental electromagnetic material properties. In particular,

it provides access to the observed SLC image $\boldsymbol{y} \in \mathcal{Y}$, an intermediate SLC image $\boldsymbol{z} \in \mathcal{Z}$ obtained before the system transfer function $\boldsymbol{H}$ is applied (corresponding to $\mathcal{C}(\boldsymbol{x}^{\frac{1}{2}}) \odot \boldsymbol{n}$ in (2)), and the scene reflectivity $\boldsymbol{x} \in \mathcal{X}$. Although $\boldsymbol{x}$ does not represent a physically measurable quantity, it serves as an effective GT candidate because it is easily interpretable by human operators. The different images are illustrated in Figure 1.

(a) Scene reflectivity $\boldsymbol{x}$. (b) Amplitude of SLC $\boldsymbol{z}$. (c) Amplitude of SLC $\boldsymbol{y}$.

Fig. 1: Images produced with MOCEM.

The MOCEM software also provides, for each scene, detailed metadata $\boldsymbol{m} = [m_1, \cdots, m_d]^\top \in \mathcal{M} \triangleq \mathbb{R}^d$, where d is the number of metadata parameters per scene, including radar geometry, squint angle, acquisition resolution, noise level, and other acquisition-specific parameters. This supplementary information can be leveraged to further enhance model performance.

Our approach is to train restoration NNs exclusively on simulated SAR images generated by MOCEM, taking advantage of both the known GT reflectivity scenes and the associated metadata. Once trained, these models are directly applied to real SAR acquisitions for inference, enabling image restoration despite the absence of paired real data. This Sim2Real strategy bridges the gap between synthetic training and operational deployment by exploiting the physical fidelity of simulation to generalize effectively to real-world SAR imaging scenarios.

C. Input Variations and Custom Loss Functions for Improved Performance

For the supervised learning task, the minimization problem (4) for training the NN $f_{\boldsymbol{\theta}}$ can be generalized as

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{v},\boldsymbol{w})\sim p(\boldsymbol{v},\boldsymbol{w})}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{v}), \boldsymbol{w})] \tag{5}$$

where $\boldsymbol{v} \in \mathcal{V}$ is the input of the NN, $\boldsymbol{w} \in \mathcal{W}$ is the target and $p(\boldsymbol{v}, \boldsymbol{w})$ is the joint distribution. Defining the set of possible inputs, targets, loss functions, and model architectures can have a significant impact on restoration performance. As a model choice, considering the effectiveness of the DRUNet architecture [26], originally designed for Gaussian denoising in a PnP ADMM framework, we chose to retain it and adapt it for SAR image restoration. We also implemented a DRUNet enhanced with Squeeze-and-Excitation (SE) blocks—modules that perform a squeeze operation via global pooling followed by an multilayer perceptron (MLP) generating channel-wise excitation weights to recalibrate feature maps [27]—which we call SEDRUNet. We retain $\boldsymbol{v} = (\tilde{\boldsymbol{y}}, \boldsymbol{y}^{\text{re}}, \boldsymbol{y}^{\text{im}})$, stacking the amplitude $\tilde{\boldsymbol{y}}$ alongside both the real and imaginary parts, $\boldsymbol{y}^{\text{re}}$ and $\boldsymbol{y}^{\text{im}}$, as the input of the NNs. Although the amplitude information is contained in the real and imaginary parts of $\boldsymbol{y}$, we experimented (results not shown in the paper) and found that using information from these three channels improves performance. The target is set to $\boldsymbol{w} = \boldsymbol{x}$. We propose training the DRUNet architectures using different loss functions ℓ: the mean absolute error (MAE), commonly used in image restoration tasks; the perceptual loss (PL) [28], which compares high-level feature representations extracted from a pretrained network to preserve perceptual quality; and the edge preserving loss (EPL) [29], which emphasizes the retention of sharp edges and fine structures by penalizing differences in image gradients. For the following, we denote by DRUNet EPL (resp. DRUNet PL) the models trained using the corresponding loss function.

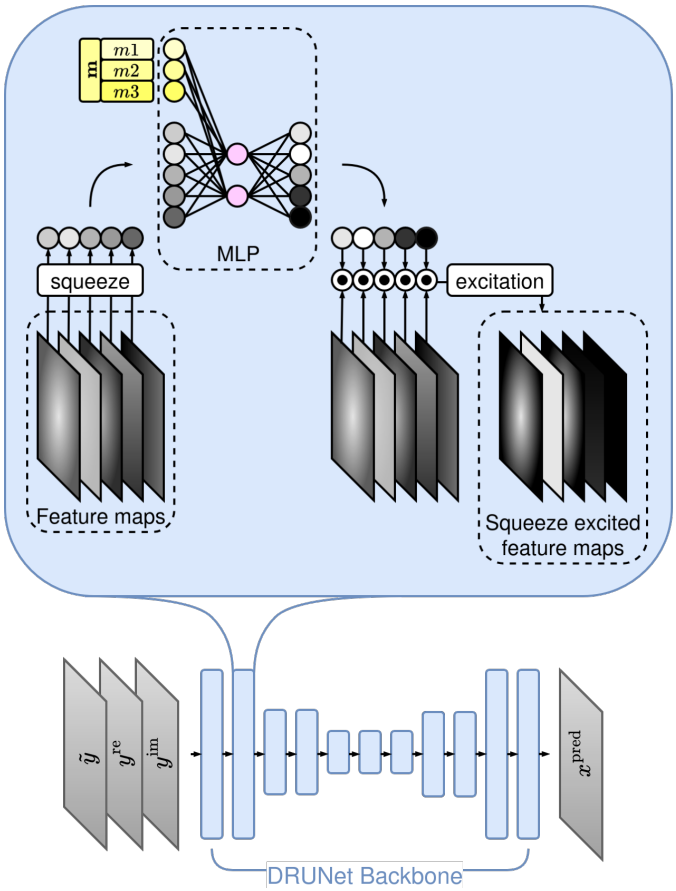


Fig. 2: Overview of the architecture of M-SEDRUNet.

Finally, we also leveraged the available metadata in the dataset to enhance the model's restoration. In that context, $\boldsymbol{v} = (\tilde{\boldsymbol{y}}, \boldsymbol{y}^{\text{re}}, \boldsymbol{y}^{\text{im}}, \boldsymbol{m})$ and $\boldsymbol{w} = \boldsymbol{x}$ remain the same as before. We propose two different methods to incorporate the metadata. The naive one, which we call M-DRUNet, consists of passing the metadata through input maps, similar to image metadata maps. For each metadata parameter m_i, a map of the same spatial dimension as the input amplitude $\tilde{\boldsymbol{y}}$ is created

and filled with the corresponding single metadata value. The second approach, called M-SEDRUNet, incorporates metadata through SE blocks, inspired by the work of Plutenko et al. [30]. Metadata are then injected into the MLPs of the SE blocks, enabling it to adapt dynamically to context. Figure 2 shows the metadata injection principle for M-SEDRUNet.

III. Experiments

A. Data Preparation

As previously mentioned, we used a dataset of SAR images simulated using the MOCEM software. The GT images of size $n = 256 \times 256$ are composed of background maps into which various CAD models are inserted. To ensure scene diversity, a total of 78 distinct CAD models—representing both military and civilian objects such as airplanes, boats, antennas, land vehicles, and buildings—are randomly placed with varying scales and orientations onto background images. The background images themselves depict a range of environments, including forests, roads, fields, and airports. From these composite scenes, MOCEM is used to simulate realistic SAR acquisitions. It is important to note that, in this case, MOCEM accurately models the electromagnetic scattering behavior only for the CAD objects; the background images do not reflect true radar backscattering properties and are included solely to enhance contextual realism. To ensure diversity in acquisition scenarios, the simulation process involves uniform sampling of scene and sensor parameters, including radar acquisition settings such as squint angle, spatial resolution, and noise level. This variability contributes to the generation of a rich and diverse dataset, better preparing the trained NNs for real-world generalization. The final generated dataset consists of 5,000 4-tuples $(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}, \boldsymbol{m}) \in \mathcal{Y} \times \mathcal{Z} \times \mathcal{X} \times \mathcal{M}$, which were split into 4,631 training samples and 369 validation samples. The split was performed by isolating specific acquisition parameter ranges, as shown in Table I, to ensure that the validation set contains conditions not present in the training set. This design explicitly tests the NNs ability to generalize to unseen radar acquisition configurations. Note that all images are standardized according to the statistics of the training split. The same standardization procedure is applied individually to each metadata variable.

Metadata \ Dataset	Training set	Validation set
Bearing (°)	$[0, 150[\cup]155, 360]$	$[150, 155]$
Incidence (°)	$[15, 55[\cup]55.7, 75]$	$[55, 55.7]$
Squint (°)	$[0, 15[\cup]15.5, 45]$	$[15, 15.5]$
Resolution (m^2)	$[0.2, 0.35[\cup]0.36, 0.6]$	$[0.35, 0.36]$
Noise level (db)	$[-40, -32[\cup] - 31.7, -20]$	$[-32, -31.7]$

TABLE I: Partitioning of the training and validation sets based on non-overlapping metadata intervals of simulated acquisitions.

Finally, we used real UMBRA[1] and CAPELLA[2] SLC images as a test set. These images have statistics properties similar to those seen in training, thereby reducing the domain gap.

[1]https://umbra.space/open-data/

[2]https://www.capellaspace.com/earth-observation/gallery

B. Experimental Settings

For performance evaluation, we compare our proposed methods specifically with SARCAM [31], a deep learning architecture designed for SAR image despeckling, as well as with AdaIR [32] and IRNeXt [33], originally developed for general image restoration. They are included in our comparison as they are well suited to the joint SAR despeckling and sidelobes reduction problem. In addition, we consider MERLIN [22] and MuLoG-DRUNet [24], well-established and effective approaches for real SAR data despeckling. Their inclusion allows us to benchmark our method against robust despeckling baselines, even though they do not explicitly target sidelobes reduction. Note that both models are evaluated only on real SAR images using publicly available pretrained weights.

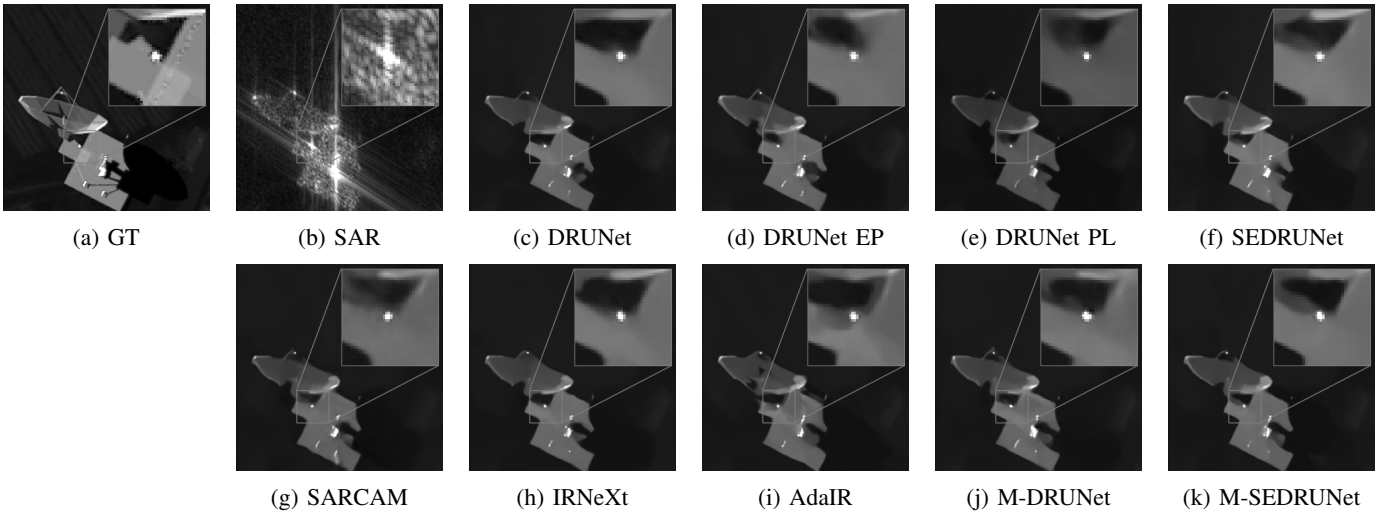
All models are trained using the Adam optimizer with an initial learning rate of 1×10^{-4}, coupled with an exponential decay schedule. Training is conducted for 200 to 400 epochs, depending on when convergence is observed on the validation set. The batch size is adjusted according to available GPU memory. To improve generalization, data augmentation techniques such as random flips and rotations are applied. We use classical image restoration metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to evaluate image quality. In addition, we include feature similarity index (FSIM) [34] and learned perceptual image patch similarity (LPIPS) [35], which capture perceptual fidelity and better align with human visual perception. To evaluate despeckling performance on real SAR images, we use the equivalent number of looks (ENL), which measures the level of speckle noise reduction in homogeneous SAR image regions. No quantitative metric for sidelobes reduction is used, as no established metric currently exists.

C. Results on Simulated SAR Images

Model \ Metric	PSNR ↑	SSIM ↑	FSIM ↑	LPIPS ↓
SARCAM	29.915	0.897	0.877	0.347
IRNeXt	30.023	0.899	0.874	0.342
AdaIR	30.040	0.899	0.878	**0.333**
DRUNet	29.794	0.897	0.876	0.338
SEDRUNet	30.012	0.899	0.874	0.339
DRUNet PL	29.919	0.897	0.879	0.341
DRUNet EPL	29.836	0.896	**0.880**	0.340
M-DRUNet	30.272	0.899	0.878	0.339
M-SEDRUNet	**30.461**	**0.901**	0.874	0.335

TABLE II: Quantitative evaluation on the simulated SAR validation set.

In terms of PSNR, the best-performing models are those incorporating metadata, which highlights the impact of metadata integration on the image restoration task. On the other hand, according to perceptual metrics such as SSIM, FSIM,



(a) GT (b) SAR (c) DRUNet (d) DRUNet EP (e) DRUNet PL (f) SEDRUNet

(g) SARCAM (h) IRNeXt (i) AdaIR (j) M-DRUNet (k) M-SEDRUNet

Fig. 3: GT and and restored simulated SAR images.

and LPIPS, all models demonstrate comparable performance. The comparison between DRUNet and SEDRUNet and their metadata-driven counterparts demonstrates that metadata injection enhances model performance, even though the improvement margin remains modest. From a visual standpoint, the restored images appear similar across all models (see Figure 3). Speckle noise is completely suppressed, and sidelobes are clearly eliminated in all restored outputs.

D. Results on Real SAR Images

Table III summarizes the quantitative evaluation of all models on real SAR imagery using the ENL metric. The despeckling-oriented methods MERLIN and MuLoG-DRUNet exhibit strong performance, with MuLoG-DRUNet achieving the best overall ENL, confirming their effectiveness on real data. The metadata-driven M-SEDRUNet attains the second-highest ENL, surpassing all DRUNet-based baselines (DRUNet, DRUNet EPL, DRUNet PL) and demonstrating the benefit of incorporating acquisition metadata for improved generalization to real imagery. IRNeXt and AdaIR also yield competitive results, indicating that general-purpose restoration models can adapt well to SAR despeckling. Overall, these results highlight that metadata-aware architectures can approach the performance of specialized despeckling methods while simultaneously addressing sidelobes reduction.

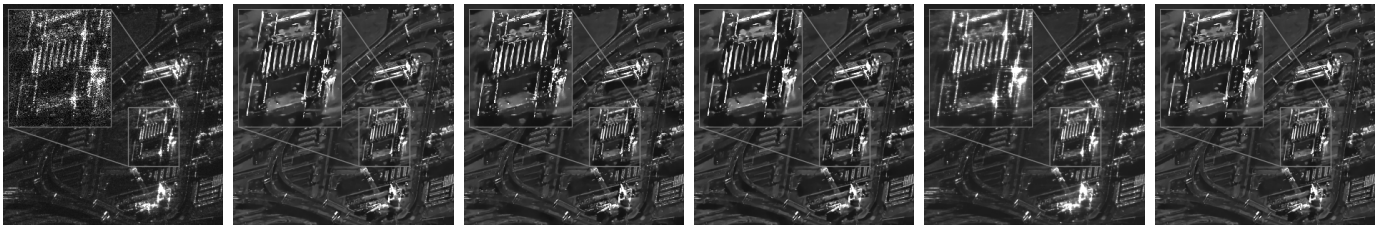
Model \ Metric	ENL ↑
MERLIN	257.6
MuLoG DRUNet	**367.7**
SARCAM	121.9
IRNeXt	242.1
AdaIR	173.1
DRUNet	83.35
SEDRUNet	175.7
DRUNet PL	73.71
DRUNet EPL	60.15
M-DRUNet	121.2
M-SEDRUNet	286.2

TABLE III: Quantitative evaluation on real SAR images.

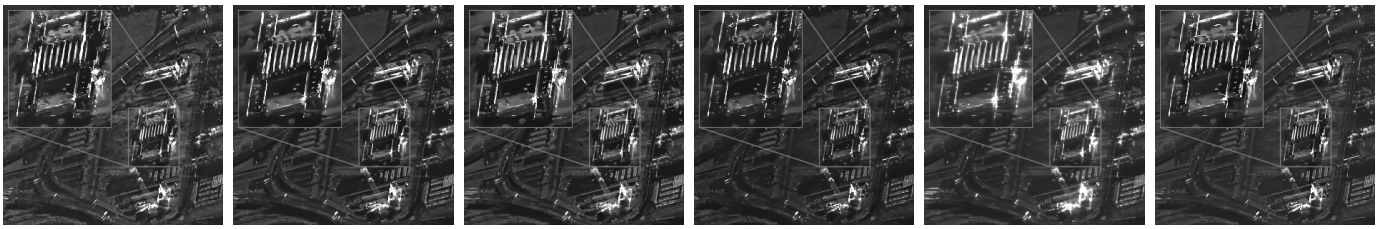
While ENL does not capture other critical aspects of SAR image restoration, such as the suppression of sidelobes or detail preservation in structured areas, relying solely on ENL may provide an incomplete assessment of a model's performance. Visual inspection therefore remains crucial for evaluating the overall quality of SAR image restoration.

Regarding the results shown in Figure 4, the despeckling-only methods MERLIN and MuLoG-DRUNet exhibit strong speckle suppression performance on real SAR images but fail to remove sidelobe artefacts, as they are not designed for joint restoration. In contrast, the proposed approaches, along with general-purpose models such as IRNeXt and AdaIR, demonstrate both despeckling and sidelobes reduction capabilities. While most methods effectively smooth homogeneous regions, IRNeXt tends to over-smooth certain areas, achieving strong sidelobes suppression at the cost of texture fidelity. DRUNet-based models generally preserve more structural details, though some loss is observed in regions with repetitive patterns, particularly for DRUNet PL. The metadata-enhanced variants maintain restoration quality comparable to their non-metadata counterparts, offering a balanced trade-off between noise suppression and detail preservation. Additional visual results are presented in Figure 5.

Even if the effects of metadata are not always directly observable in the restored images, an interesting outcome is that metadata-driven models can implicitly interpret the underlying physics of the acquisition parameters for the restoration task. For example, injecting a resolution parameter that differs from the one used by the sensor to acquire the original SAR image can influence the restored image's effective resolution. Figure 6 clearly illustrates that providing a higher resolution parameter leads to visibly enhanced image detail. This capability, while promising, has some drawbacks. In particular, it can introduce new sidelobes or artefacts when the injected metadata significantly deviate from the original acquisition parameters. Nonetheless, this adaptive functionality opens new possibilities for dynamic image analysis and restoration tuning. Similar behaviors are observed when varying the incidence angle.

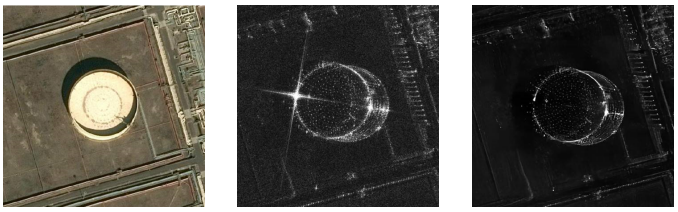


(a) SAR (b) DRUNet (c) DRUNet EPL (d) DRUNet PL (e) MERLIN (f) SEDRUNet



(g) SARCAM (h) IRNeXt (i) AdaIR (j) M-DRUNet (k) MuLoG DRUNet (l) M-SEDRUNet

Fig. 4: Restored images from real UMBRA SLC.



(a) (b) (c)

Fig. 5: Optical GT (a) alongside CAPELLA SAR image (b) and restored image with SEDRUNet (c).

(a) (b) (c)

Fig. 6: UMBRA SAR image (a) and restorations with M-SEDRUNet using the real sensor resolution set to 0.40m^2 (b) and a different one set to 0.25m^2 (c).

IV. Discussion

This study demonstrates the potential of the Sim2Real approach, as well as the integration of SAR acquisition parameters into the NNs to enhance performance. However, several limitations must still be addressed.

The injection of metadata into DRUNet and SEDRUNet has shown improved performance, particularly in simulated environments where GT data are accessible. Comparisons using identical architectural baselines confirm the benefits of this approach. However, we have not yet explored metadata injection in other advanced models such as AdaIR and IRNeXt. Integrating metadata into these architectures could potentially further enhance their performance and remains an avenue for future investigation.

While the current results confirm that metadata can guide the restoration process by modulating the network's behavior according to acquisition conditions, the quantitative characterization of this control remains limited. Specifically, the trade-off between the degree of metadata influence and the risk of artefact introduction has not yet been systematically evaluated. Future work should therefore investigate how varying the weighting or conditioning strength of metadata impacts quantitative metrics and perceptual fidelity.

One major challenge is the domain gap between the simulated images used during training and real SAR images. For certain SAR image providers, such as TerraSAR-X[3], the statistical properties of real images deviate from those seen in training, leading to poor image restoration results. Currently, no effective solution has been identified to reconcile these statistical discrepancies, highlighting the limitations of supervised training strategies in this context. This motivates the exploration of unsupervised approaches. Recently, deep posterior sampling (DPS) approaches [36] have gained attention for solving inverse problems, where prior information is learned using diffusion models. Methods such as BlindDPS [37] aim to jointly estimate the forward operator and the latent clean image $\boldsymbol{x}$. Applying such techniques in the SAR domain may enable the joint estimation of both the SAR transfer function and the underlying image, offering a promising direction for future work.

V. Conclusion

In this work, we presented a unified deep learning framework for simultaneous despeckling and sidelobes reduction in SAR images. Leveraging the realistic SAR simulated dataset

[3] https://earth.esa.int/eogateway/missions/terrasar-x-and-tandem-x/sample-data

generated by MOCEM, which includes access to GT, we enabled supervised training that effectively addresses both inherent noise and artefacts in SAR imagery. Our results demonstrate that incorporating acquisition metadata as auxiliary input improves restoration performance and enables the guided control of image restoration.

ACKNOWLEDGMENTS

This study has been carried out with financial support from the French Direction Générale de l'Armement and the French Agence Ministérielle pour l'Intelligence Artificielle de Défense.

REFERENCES

[1] A. Nuttall. Some windows with very good sidelobe behavior. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(1):84–91, 1981.

[2] H.C. Stankwitz, R.J. Dallaire, and J.R. Fienup. Nonlinear apodization for sidelobe control in sar imagery. *IEEE Transactions on Aerospace and Electronic Systems*, 31(1):267–279, 1995.

[3] Rémy Abergel, Loïc Denis, Saïd Ladjal, and Florence Tupin. Subpixellic methods for sidelobes suppression and strong targets extraction in single look complex sar images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):759–776, 2018.

[4] Sen Yuan, Ze Yu, Chunsheng Li, and Shusen Wang. A novel sar sidelobe suppression method based on cnn. *IEEE Geoscience and Remote Sensing Letters*, 18(1):132–136, 2021.

[5] Shuyi Liu, Yan Jia, Yongqing Liu, Limin Zhai, and Xiangkun Zhang. A new birnn-sva method for side lobe suppression. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:1167–1175, 2024.

[6] J. W. Goodman. *Laser Speckle and Related Phenomena.*, chapter Statistical properties of laser speckle patterns. Springer Berlin., 1984.

[7] José M. Bioucas-Dias and Mário A. T. Figueiredo. Multiplicative noise removal using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(7):1720–1730, 2010.

[8] Charles-Alban Deledalle, Loic Denis, Sonia Tabti, and Florence Tupin. Mulog, or how to apply gaussian denoisers to multi-channel sar speckle reduction? *IEEE Transactions on Image Processing*, 26(9):4389–4403, September 2017.

[9] G. Chierchia, D. Cozzolino, G. Poggi, and L. Verdoliva. Sar image despeckling through convolutional neural networks. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5438–5441, 2017.

[10] Puyang Wang, He Zhang, and Vishal M. Patel. Sar image despeckling using a convolutional neural network. *IEEE Signal Processing Letters*, 24(12):1763–1767, 2017.

[11] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[13] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa E. Alarcón. A residual dense u-net neural network for image denoising. *IEEE Access*, 9:31742–31754, 2021.

[14] Fan Jia, Wing Hong Wong, and Tieyong Zeng. Ddunet: Dense dense u-net with applications in image denoising. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 354–364, 2021.

[15] Chi-Mao Fan, Tsung-Jung Liu, and Kuan-Hsien Liu. Sunet: Swin transformer unet for image denoising. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2022.

[16] Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation, 2018.

[17] Jaekyun Ko and Sanghwan Lee. Sar image despeckling using continuous attention module. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:3–19, 2022.

[18] Puyang Wang, He Zhang, and Vishal M. Patel. Generative adversarial network-based restoration of speckled sar images. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5, 2017.

[19] Ruijiao Liu, Yangyang Li, and Licheng Jiao. Sar image specle reduction based on a generative adversarial network. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2020.

[20] Malsha V. Perera, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M. Patel. Sar despeckling using a denoising diffusion probabilistic model. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023.

[21] Xuran Hu, Ziqiang Xu, Zhihan Chen, Zhengpeng Feng, Mingzhe Zhu, and LJubisa Stankovic. Sar despeckling via regional denoising diffusion probabilistic model, 2024.

[22] Emanuele Dalsasso, Loïc Denis, and Florence Tupin. As if by magic: Self-supervised training of deep despeckling networks with merlin. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[23] Liang Chen, Yifei Yin, Hao Shi, Qingqing Sheng, and Wei Li. A self-supervised sar image despeckling strategy based on parameter-sharing convolutional neural networks. 2023.

[24] Cristiano Ulondu Mendes, Loïc Denis, Charles Deledalle,

and Florence Tupin. Robustness to spatially-correlated speckle in plug-and-play polsar despeckling. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[25] Christian COCHIN, Philippe POULIGUEN, Benoit DELAHAYE, Daniel le HELLARD, Philippe GOSSELIN, and Franck AUBINEAU. Mocem - an 'all in one' tool to simulate sar image. In *7th European Conference on Synthetic Aperture Radar*, pages 1–4, 2008.

[26] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior, 2021.

[27] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[28] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[29] Ram Krishna Pandey, Nabagata Saha, Samarjit Karmakar, and AG Ramakrishnan. Msce: An edge-preserving robust loss function for improving super-resolution algorithms. In *International Conference on Neural Information Processing*, pages 566–575. Springer, 2018.

[30] Iaroslav Plutenko, Mikhail Papkov, Kaupo Palo, Leopold Parts, and Dmytro Fishman. Metadata improves segmentation through multitasking elicitation. In *MICCAI Workshop on Domain Adaptation and Representation Transfer*, pages 147–155. Springer, 2023.

[31] Jaekyun Ko and Sanghwan Lee. Sar image despeckling using continuous attention module. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:3–19, 2021.

[32] Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. Adair: Adaptive all-in-one image restoration via frequency mining and modulation. *arXiv preprint arXiv:2403.14614*, 2024.

[33] Yuning Cui, Wenqi Ren, Sining Yang, Xiaochun Cao, and Alois Knoll. Irnext: Rethinking convolutional network design for image restoration. 2023.

[34] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011.

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[36] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[37] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6059–6069, 2023.

Physics-inspired data augmentation for SAR ATR: a new approach to tackle the synthetic-to-measured Domain Gap

Elisa Delhommé
Thales LAS France SAS
Elancourt, France
0009-0007-9501-2117

Héloïse Remusati
Thales LAS France SAS
Elancourt, France
0000-0002-5528-5491

Caroline Lesueur
Thales LAS France SAS
Elancourt, France
0009-0000-8557-7180

Jacques Petit-Frère
Thales LAS France SAS
Elancourt, France
0009-0008-8384-5581

***Abstract*—Synthetic Aperture Radar (SAR) imaging supports applications from environmental monitoring to defence. For Automatic Target Recognition (ATR), Deep Learning (DL) delivers strong results but needs large datasets, and SAR data is scarce due to cost and confidentiality. A common workaround is training on synthetic data generated by simulators and Computer-Aided Design (CAD) models, but these simplify complex electromagnetic effects, creating a domain shift between training (synthetic) and test (measured) domains. Although Data Augmentation (DA) is used to improve representativeness and robustness, many methods lack semantic, physics-informed changes to improve recognition performance. In this paper, we propose a physics-based DA to address the Synthetic-to-Measured (S2M) gap, first validating physical parameter extraction from measured images and then leveraging this knowledge to improve ATR. Training solely on synthetic data, our approach achieves 70.97% accuracy.**

***Index Terms*—ATR, classification, synthetic data, MOCEM, SAR, data augmentation, ASC, deep learning, MSTAR.**

I. Introduction

Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) is often used in many real-world applications where object identification is required via a classification system. Deep learning-based SAR ATR has achieved increasingly high performance in the past few years, particularly on the open Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset under standard operating conditions (SOCs). As a matter of fact, deep neural networks have proven indispensable for tackling the complex task of classifying SAR images due to the multiple electromagnetic effects they exhibit.

Most of the studies generally show cases where the test conditions are close to the learning conditions using measured data from neighbouring depression angles (15° versus 17° most of the time). This case, although effective, is rarely adapted to real operational conditions where exact target information is rare or even unavailable, and where learning on synthetic data is recommended (or even required). However, synthetic data are based on simulators that cannot accurately reproduce all the effects present in the measured data and cannot cover all the variability found under real operating conditions. As a result, models trained on these data often have limited generalisation and robustness potential in the real world. This is why data augmentation (DA) techniques are essential, whether to bring synthetic and real data closer together, to present numerous variations to the model, or simply to expand the training database. SAR images although have unique physical properties that set them apart from optical images, for which most data augmentation techniques in the literature are designed and which do not always make physical sense for this type of data (for example, rotations or affine transformations). Techniques designed specifically for the optical domain may raise questions, such as:

When does a target cease to be the same? While this question is more obvious in optics, it is more difficult to answer when dealing with a SAR image to which transformations are to be applied. Some transformations, such as affine transformations, can change the target shape.

How to ensure that the applied transformations are sufficient to cover the target domain? This kind of augmentation techniques can improve the extraction of features of interest for classification and provide generalisation ability for data exhibiting the same type of variability. However, this does not guarantee robustness to other variations such as pixel level modification or target state variation (open doors...).

This paper therefore introduces a data augmentation technique that addresses three issues:

- Improving recognition performance while learning only from synthetic data,
- Providing a new data augmentation technique for SAR images,
- Introducing an artificial augmentation that resorts to physical mechanisms and that is interpretable.

The proposed SAR augmentation process draws on physical knowledge based on scattering mechanisms. The objective of this method is to confront synthetic training databases with real measurements in order to achieve better performance and robustness of SAR ATR algorithms. We believe that the use of physics-based transformations can improve the guarantee that the algorithm will work in real-world conditions and cover the domain of use, by providing interpretable augmented samples. The paper is presented as follows. Section II introduces background and related works. Section III brings in our approach

and presents the methodology adopted for carrying out this work. Section IV shows the results of the data analysis and details the results obtained for SAR ATR on the MSTAR dataset. Section V addresses the limitations of the proposed method. Finally, Section VI concludes the paper and discusses future work.

II. Related works

When training machine learning algorithms, it is typically assumed that the data distribution of the training and test sets is consistent. Nevertheless, this assumption often fails to hold in practical applications and in particular when training is performed on synthetic data [3]. Indeed, even though simulation models are particularly efficient in providing physically accurate SAR images, they are only simplified representations of the complex mechanisms that form the signature of targets in images [2], leading to residual discrepancies between measured and synthetic images. For example, Fig. 1 illustrates a visible domain gap between synthetic and real images.

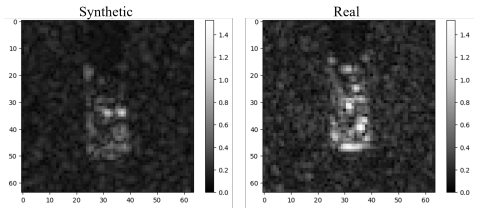


Fig. 1. Domain Gap Example: 2S1 target (azimuth 358°). Images are displayed with the same colour range with the QPM LUT.

Domain generalisation and domain adaptation are areas of research aiming at alleviating this issue, often referred to as domain shift or gap. To do so, several strategies have been proposed in the literature to make models able to generalise, to an unseen domain (also called target domain – measured samples in our case), using a known domain (also called source domain – synthetic samples in our case). They can be primarily categorised into three types [3], being domain alignment, meta-learning, and data augmentation, but we are going to focus on the last one in this paper. More specifically, data augmentation is the process of artificially generating new data from existing data.

Particularly for the SAR S2M domain gap, we can name two different (non-exhaustive) data augmentation strategies:

Image-level augmentation. According to [3], it creates new examples similar to the original images but does not explicitly focus on the concept of domain by performing a series of transformations or noise addition to the original images in the training set. For example, in [5], Inkawhich et al. tested and compared several data augmentation techniques such as rotation and the addition of Gaussian noise and analysed their impact on saliency and feature-space representation. Similarly, in his PhD work, Denton [6] proposed an alpha blending approach to generate new data points by combining SAMPLE extracted targets with MSTAR clutter samples.

Domain-level augmentation. According to [3], this type of data augmentation method tries to increase the breadth of the training domains and to cover the unseen target domain as comprehensively as possible by generating a large number of new samples with diverse distributions. For example, the authors of [4] directly worked on synthetic and measured images in order to bring them closer. They observed that the synthetic and measured images in the SAMPLE dataset are linearly separable and utilise a linear content erasure method (LCDE) in order to transform the images and eliminate this separability. Differently, Camus et al. in [2] proposed a domain randomisation technique, in order to introduce randomness and variations into samples within a simulated environment, combined with adversarial training. More recently, they proposed in [12] an even more advanced image generation pipeline, by combining their ADASCA block, presented in [2], with two other kinds of domain-level augmentation: semantic and radar augmentations using, respectively, their ADAMO and M3D Exploit tools.

Similarly to our approach, these different works try to bridge the domain gap between synthetic and measured SAR images. Nevertheless, we realise that most research papers:

1) either try to characterise and bridge the domain gap by using learnt transformations (for example [4]), but do not provide physical comprehension of them (for instance, synthetic and measured samples exhibit specific target signature differences),
2) or produce new samples without any guidance, hoping that the augmented database will cover the target domain as we assume that more diverse data can help reduce generalization error (for example [12]).

As a consequence, and as stated in [3], we believe that addressing the following question is of utmost importance: "which data augmentation method is most useful in the current task and why". Indeed, currently, most data augmentation methods create samples that are not interpretable, such as adversarial perturbation and adding noise, or without being able to justify why they will be beneficial. With our approach, we propose to resort to physical knowledge provided by Attributed Scattering Centres (ASC) in order to both bridge the gap between synthetic and measured samples and physically characterise it. We believe that our approach, combined with other DA techniques, such as the one of SCALIAN DS [12], can improve even more recognition performance.

By passing, we found some papers in the literature that also used attributed scattering centres for data augmentation (for instance [7], [8]), but they only consider the measured-measured configuration and do not deal with the synthetic-measured gap. To our knowledge, this is the first work to attempt to benefit from ASC to characterise and bridge the synthetic-to-measured domain gap.

III. Methodology

Our approach uses a physical modelling of SAR images, provided by the Attributed Scattering Centres Model. Thanks to it, it is possible to describe both synthetic and measured

images and to use this physical knowledge to generate augmented samples. Even though similar to the pipeline proposed by SCALIAN DS in [12], our modelling and proposed DA technique differs on two aspects: physical parameters can be estimated from both synthetic and measured images, without being adherent to MOCEM or CAD models, and can thus provide an interpretable feature space shared between the two domains, providing hints about their discrepancies.

A. Our approach

1) Attributed Scattering Centres (ASC) model: Proposed by Gerry et al. [9] in 1997, the Attributed Scattering Centres (ASC) model is based on the geometrical theory of diffraction (GTD) and physical optics (PO) and enables one to accurately model the scattering of a target. This model deals with both localized (the scatterer appears to exist at a single point in space) and distributed (the scatterer, in the imaging plane, appears as a finite, non-zero-length current distribution) scattering mechanisms, and characterises them with a set of several parameters corresponding to:

- Frequency and aspect dependence,
- Physical attributes (such as the structure, location, orientation, geometry, size).

Formally, the ASCM describes the total scattered field as a function of frequency f and aspect angle ϕ as:

$$E(f,\phi;\Theta_N) = \sum_{i=1}^{N} E_i(f,\phi;\theta_i) \tag{1}$$

Where $\Theta_N = \{\theta_i | \theta_i = [A_i, x_i, y_i, \alpha_i, \gamma_i, L_i, \bar{\phi}_i], 1 \leq i \leq N\}$ is the parameter set of N individual scatterers and

$$\begin{aligned} E_i(f,\phi;\theta_i) =& A_i \cdot (j\frac{f}{f_c})^{\alpha_i} \\ & \cdot \exp(-j\frac{4\pi f}{c}(x_i\cos\phi + y_i\sin\phi)) \\ & \cdot \text{sinc}(2\pi f c L_i \sin(\phi - \bar{\phi}_i)) \\ & \cdot \exp(-2\pi f \gamma_i \sin\phi) \end{aligned} \tag{2}$$

With each term $E_i(f,\phi;\theta_i)$ representing the backscatter from a single scattering mechanism, f_c the centre frequency of radar wave, c the velocity of light.

The parameter set of the ith scatterer has physical interpretations related to the location and the geometry of the scatterer. Specifically, A_i represents the relative amplitude of the measured field ($A_i \in \mathbb{C}^2$), x_i and y_i correspond, respectively, to range and cross-range locations (in meters), $\alpha_i \in \{-1, -0.5, 0, 0.5, 1\}$ models the frequency dependence, γ_i describes the mild aspect dependence of localized scattering centre cross section, L_i models the length of the scattering centre and $\bar{\phi}_i$ models the orientation angle with respect to the broadside.

This model has the advantage of being able to represent several types of scattering centres depending on the different values the parameters take, as presented in Table I.

TABLE I
GEOMETRIC SCATTERING TYPES DIFFERENTIATED BY FREQUENCY AND ASPECT DEPENDENCE.

	Geometric Scattering Type	α	γ	L	ϕ
Localized	Trihedral	1	> 0	0	0
	Top Hat	0.5			
	Sphere	0			
	Corner Diffraction	-1			
Distributed	Dihedral	1	0	> 0	$\neq 0$
	Cylinder	0.5			
	Edge Broadside	0			
	Edge diffraction	-0.5			

2) Extracting ASC parameters: Given a SAR image $D(x,y)$, the objective is to find the set of parameters $\Theta_N = [\theta_0, ..., \theta_N]$ that best fit the N scattering mechanisms present in the object that is imaged.

To do so, we resort to the minimisation of a cost function that represents the difference between the model and the actual response in the image domain, as scattering centre responses are isolated in the image domain when data is gathered at high frequencies. Consequently, our problem can be expressed as:

$$\hat{\Theta}_N = \arg\min_{\Theta_N} \left\| D(x,y) - \tilde{D}(x,y;\Theta_N) \right\|_2 \tag{3}$$

where Θ_N are the scattering parameters, $D(x,y)$ the SAR image and $D(x,y) - \tilde{D}(x,y;\Theta_N)$ the reconstructed image via the ASC model.

First, as with every numerical optimisation process, we initialise a set of values before starting the iterations. Next, we use a gradient-based optimisation to numerically obtain the set of parameter values that minimise the cost function. Choosing the order of the problem, here N, that corresponds to the number of scatterers, is really difficult. For simplicity, we fix it to a upper bound, the value of which we will examine in IV-B1b. It is worth noting that this number of scatterers will change, depending on the considered target, azimuth angle and database.

3) Reconstructing images from ASC parameters: Given the set of ASC parameters, it is next possible to generate the SAR signal in the frequency-aspect angle domain using the equation 1. In order to retrieve the Cartesian domain data, it is then needed to resample to a uniform grid on coordinates (f_x, f_y) expressed as $f_x = f\ \cos(\phi)$ and $f_y = f\ \sin(\phi)$.

This resampling gives us an $M \times P$ array depending on f_x and f_y in the frequency domain. In order to be comparable with SAR images, we need to convert these data into the image domain. To do so, the data are first multiplied by a Taylor window (with a -35 dB side lobe level), and is next zero-padded to a new size of $M_z \times P_z$ where $M_z = 1.5M$ and $P_z = 1.5N$. Finally, we get the SAR image $D(x,y)$ thanks to a two-dimensional inverse Fourier (2D-IFFT). It is thus possible to compare the images generated with the ASCM with measured samples.

4) Using ASC for data augmentation:

a) Training pipeline: In the context of SAR ATR, our approach uses the ASC model to obtain a representation in

the ASC domain of sample images. In this way, the data augmentation process can be carried out in a domain ancillary to SAR images, which is more physical and allows for better control of the transformations performed.

The training pipeline relies on the process presented in Fig. 2:

1) Starting from a synthetic training dataset, the ASC parameters are extracted on each image separately, obtaining an ASC training dataset.
2) Augmentations and transformations are then optionally performed on the ASC parameters (red block).
3) In order to train the model of interest on SAR images, the reconstruction step is necessary to convert the ASC parameters back into SAR images.

We then obtain a synthetic reconstructed training dataset. An example of reconstruction is shown in Fig. 3. In this new dataset, only the targets are reconstructed (Step 3), resulting in a zero background, which is not consistent with the real test data set.

For a matter a time, we first overcome this issue and compensate for reconstruction defects by averaging each reconstructed sample with its original version (Step 4). We emphasise the fact that it is only a temporary measure until we have implemented a better ASC parameter estimation algorithm and found a process to add background information in reconstructed images. Finally, as is traditionally the case in deep learning processes, the image is standardised: the model is trained with the Quarter Power Magnitude (QPM) LUT on the images, that are then normalised and cropped to a 64×64 format. These images are then used to train our model, which is a ResNet18. For information, we trained it for 500 epochs, with early stopping (that happened at epoch 257), with batches of 32 samples. We used an Adam optimiser with a learning rate of 0.001.

b) Transformation types: As mentioned earlier, our data augmentation method do not apply directly to the training samples, in this case SAR images, but to the ASC parameters. All augmentation types are then performed on ASC parameters data type, before the reconstruction, i.e. between stages 2 and 3 in Fig. 3. Three families of transformation are used:

- Adding Gaussian noise to the parameters, on the reflectors position or amplitude for example.
- Adding new reflectors:
  - defined (type of effects and amplitude) randomly or similar to the target.
  - placed randomly or on the target area.
- Removing random reflectors.

Fig. 4 illustrates the different types of transformation in the resulting SAR reconstruction (Step 3) in Fig. 3). The SAR reconstruction from ASC parameters is also performed on the original sample without any transformation for comparison purposes.

In this paper, we then refer to data augmentation techniques applied directly to SAR images as *image augmentations* and those applied in the domain of ASC parameters as *ASC augmentations*.

B. Study of the ASC modelling

Our first experiments aim at evaluating the quality of our ASC parameter extraction algorithm and to analyse the sets of parameters extracted from our datasets.

1) Quality evaluation of the ASC parameters extraction algorithm: The purpose of this study is to analyse the impact of ASC extraction hyper-parameters on the reconstructions, and thus select the most effective set of values for the experiments. The extraction of ASC parameters is considered effective if the reconstruction of the SAR image is close to the original image. The impact of both the number of iterations and the number of scattering centres will be evaluated.

2) Analysis of estimated ASC parameters: Analysing the ASC parameters not only allows trends in the data to be identified, but also any gaps in the data to be determined. To do this, several aspects are studied:

- The number of parameters overall and per effect according to the type of target, but also the angular sectors,
- The comparison of the parameters obtained from real and synthetic data,
- The effectiveness of extraction on both types of data.

3) Metrics: To perform the experiments presented in the previous section, we will use the following metrics.

a) Image comparison: For comparing an original image I_O and its reconstruction I_R, we use the Image Correlation Coefficient metric (ICC):

$$ICC(I_{\mathrm{O}}, I_{\mathrm{R}}) = \frac{\sum_m \sum_n \left[I_{\mathrm{O}}(m,n) - \bar{I}_{\mathrm{O}}\right]\left[I_{\mathrm{R}}(m,n) - \bar{I}_{\mathrm{R}}\right]}{\sqrt{\sum_m \sum_n \left[I_{\mathrm{O}}(m,n) - \bar{I}_{\mathrm{O}}\right]^2 \sum_m \sum_n \left[I_{\mathrm{R}}(m,n) - \bar{I}_{\mathrm{R}}\right]^2}} \quad (4)$$

We also computed other metrics but, due to space considerations, we only report ICC values that appear to be fairly aligned with the human eye.

b) ASC parameters analysis: **Heatmaps.** Heatmaps are used to show data depending on two independent variables as a colour coded image plot. In this case, they represent a number of elements associated with a colour bar.
Violin plots. Violin plots are used to compare data distributions in a similar way to box plots, but with the addition of probability density information.

C. ASC data augmentation study for SAR ATR

The focus of the following experiments is on the benefits of our DA technique regarding recognition performance. In order to quantify the contribution of the proposed method for SAR ATR, several test configurations are carried out. We gradually add different ASC augmentation techniques, starting from baselines. All tested configurations are listed in Table II. Each of the three baselines has its own objective:

- The *Baseline* allows to define a starting point for performance on original SAR images, without resorting to any data augmentation.

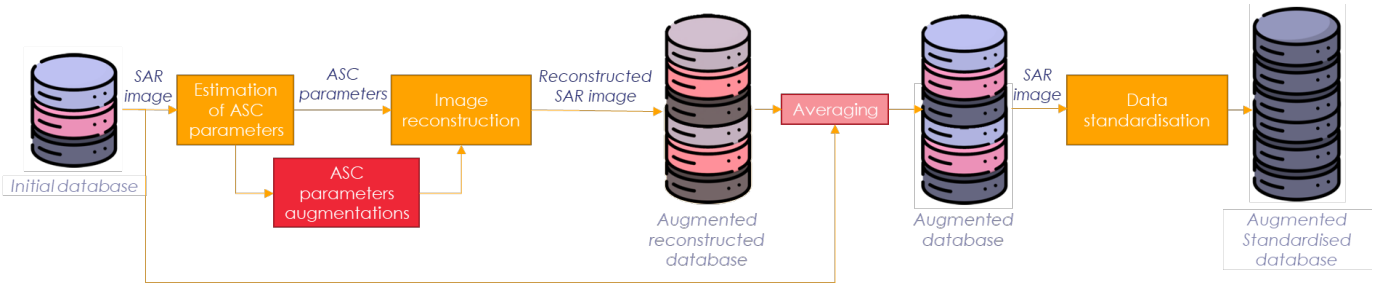


Fig. 2. Our data augmentation pipeline (ASC augmentation).

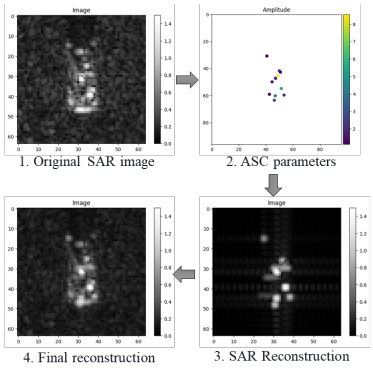


Fig. 3. Example of an ASC extraction and reconstruction on a 2S1 real target. Images are displayed with the same colour range with the QPM LUT.

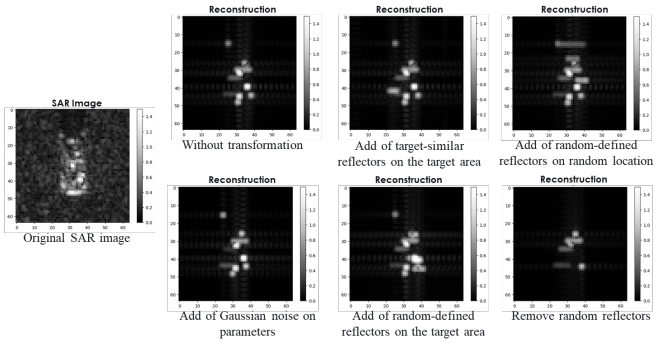


Fig. 4. The different transformations effects on the reconstructed SAR image on a real 2S1 example.
Images are displayed with the same colour range with the QPM LUT.

- The *Augmented Baseline* allows to quantify the final gain obtained by adding traditional image data augmentation techniques.
- The *Baseline ASC* represents a point of comparison for each ASC-augmented test training via SAR reconstructed images, without resorting to any data augmentation.

In order to confront our work to the state of the art, some test configurations use traditional SAR image augmentation techniques, inspired from and detailed in [1]: Gaussian noise addition, colour jittering and random area erasing. We also add random high pixels dropout and value shifting. What's more, all configurations use circular shifts (x and y offsets) on the synthetic training images (original and reconstructed) for the significant performance gain it provides, as mentioned in [2]. The study is evaluated thanks to traditional classification metrics Accuracy and F1-score.

IV. Experimental results

A. Data under study

1) Synthetic MOCEM MSTAR: The training set is based on a synthetic database generated using the MOCEM software, which is a CAD-based SAR imaging simulator developed by SCALIAN DS for the French MoD (DGA) [11]. Synthetic samples were directly provided by the French MoD who reproduced the MSTAR dataset. They used one CAD model per MSTAR class and the radar parameters provided by the MSTAR metadata. They ran parametric simulations for the depression angle of 15° and generated images at every 1° azimuth for the full [0°, 360°[range. Thus, they did not consider exactly the same azimuth angles than MSTAR. These 3600 images were randomly split into train/validation sets using a 75%/25% repartition for the first two baselines. For the *Baseline ASC* and TSM1 to TSM10, the corresponding ASC-based reconstructions are added to these splits, leading to a total of 7200 images. Special attention is paid to ensuring that both the original image and its reconstruction are included in either the training set or the validation set. Addition is considered instead of replacement to still take benefit from the initial physical information provided by MOCEM.

2) MSTAR: The test set is the measured MSTAR dataset, collected by the Sandia National Laboratory SAR sensor platform and sponsored by Defence Advanced Research Projects Agency and Air Force Research Laboratory [10]. The MSTAR dataset allows the benchmark for experiments related to problems dealing with SAR images. Each MSTAR file has the same structure: it includes experimental conditions (such as azimuth angles, depression angles, and target classes, among others), and the amplitude and phase information. The target images were captured by X-band SAR sensor and have a resolution of 0.3×0.3m. The test dataset consists of the central-cropped 64 × 64-sized 2425 SAR target images under a depression angle of 15°. This dataset contains 10 target classes that are composed of one bulldozer (D7), one truck (ZIL131), one air defence unit (ZSU234), one rocket launcher (2S1), two tanks

TABLE II
TESTING SYNTHETIC-TO-MEASURED CONFIGURATIONS

Test name	**Training dataset** (number of images)	**Augmentations**		
		ID	**Images**	**ASC**
Baseline	S[2] (3600)	-	-	-
Augmented Baseline	S (3600)	A0	Traditional SAR images augmentations[1]	-
Baseline ASC	S + R[3] (7200)	-	-	-
TSM1	S + R (7200)	A1	-	Gaussian noise on reflectors amplitude.
TSM2	S + R (7200)	A2	-	Gaussian noise on reflectors position (x and y)
TSM3	S + R (7200)	A3	-	A1 + A2
TSM4	S + R (7200)	A4	-	Adding reflectors (randomly defined and similar to the target)
TSM5	S + R (7200)	A5	-	Random removing of reflectors.
TSM6	S + R (7200)	A6	-	A3 + A4
TSM7	S + R (7200)	A7	-	A3 + A5
TSM8	S + R (7200)	A8	-	A4 + A5
TSM9	S + R (7200)	A9	-	A6 + A5
TSM10	S + R (7200)	A10	A0	A9

[1]The data augmentation methods used are detailed in III-C.
[2]S = Original synthetic dataset.
[3]R = Reconstructed synthetic dataset.

(T62, T72), and four armoured personnel carriers (BMP2, BTR60, BTR70, BRDM2).

B. Study of the ASC modelling

1) Quality evaluation of ASC extraction algorithm:

a) Impact of the number of iterations: Analysing the effectiveness of ASC parameters extraction according to different iteration values allows us to choose the optimal value for generating training data. Fig. 5 presents the ICC metrics resulting from all 2S1 synthetic SAR images compared with their pairwise reconstruction. The left plot shows sorted ICC values obtained from all pairs for different number of iterations. The right heatmap presents the same results according to azimuthal sectors. Both graphs allow us to conclude that reconstruction improves as the number of iterations increases. However, after 100 iterations, the gain on the ICC metric weakens, which is why a value of 200 seems to strike a balance between quality and generation speed.

It is also very interesting to note that ICC performs much better for certain angular ranges, particularly around 90° and 270°, which correspond to cases where the target is facing forwards or backwards, regardless of the number of iterations.

b) Impact of the number of scattering centres: To conclude on the effect of the number of scattering centres considered in the algorithm, we propose to calculate the ICC between the reconstructed image and the original image of the synthetic data for all targets. Histograms for 5, 25, 50 and 100 considered reflectors are displayed Fig. 6. Five points are clearly insufficient to reconstruct a high-quality image and a real improvement is observed from 50 points onwards. Using 100 reflectors reduces the risk of obtaining data with a low ICC, but considering the increasing calculation time with the number of reflectors, a value of 50 reflectors will be retained for the rest of the experiments.

2) Analysis of estimated ASC parameters:

a) Stability of extraction according to classes and angular sectors: To observe trends in ASC parameters in terms of number of points between different targets, but also according

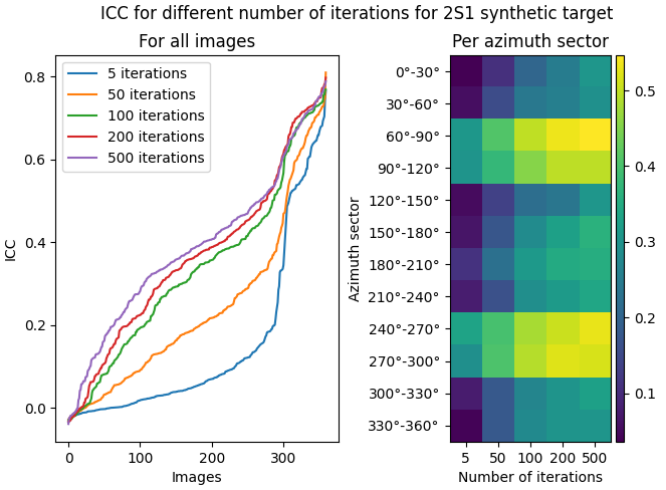


Fig. 5. ICC analysis between the original SAR image and the image reconstructed from the ASC parameters extracted according to 5, 50, 100, 200 and 500 iterations. The extraction is performed with 50 reflectors. On the left are the ICC curves sorted for all images, and on the right is the heatmap by angular sector for the different iteration values.

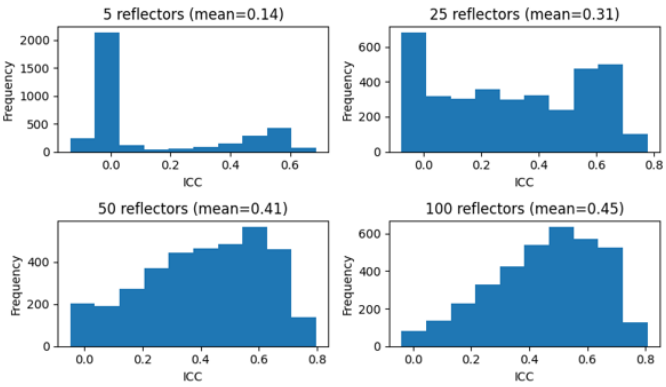


Fig. 6. ICC distribution between reconstructed and synthetic original images for 5, 25, 50 and 100 scattering scatters. The extraction is performed with 50 reflectors and 200 iterations.

to the angular sector considered, the Fig. 7. presents heat maps for synthetic and real datasets. All data is obtained by extracting 50 reflectors, but some are null. Here, we consider the number of points to be the number of non-null points among the 50.

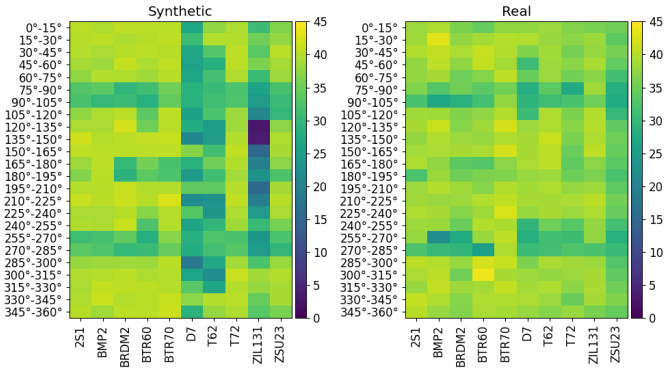


Fig. 7. Average number of reflectors per angular sector for each target for synthetic and real datasets.

Two main findings emerge from this study. Firstly, the study reveals a similar trend between synthetic and real datasets according to angular sectors around azimuths 90° and 270° which have fewer reflectors. This corresponds to cases where the target is facing forwards and backwards. This is not a problem, as it has been observed Fig. 5 that reconstruction is better at these angles.

Secondly, some synthetic targets show a significant gap with measured data, which manifests itself in a lack of reflectors. In particular, the D7 target has fewer reflectors in the synthetic case, which is consistent with its smaller size compared to other targets, but this trend, is in fact, absent in the real case. This is also the case for T62 target, even though it is similar to the T72. Also, the ZIL131 seems to be pathological for azimuths between 120° and 150° as there are practically no extracted reflectors.

The study can be refined by looking at the number of reflectors per radar effect in the same cases. Fig. 8 presenting the average number of effects for each angular sector for all targets combined shows that trihedral and corner-diffraction effects are predominant, while cylinder effects are rarely detected.

Fig. 9 focuses on the ZIL131 pathological case via a synthetic versus measured violin plot confronted to the 2S1 case. The 2S1 distributions are more symmetrical between synthetic and real cases, as shown by the EMD coefficient, which shows greater similarity than the ZIL131, which highlights significant asymmetry.

For further analysis, Fig. 10 reveals that pathological cases generally happen when a pixel or a small group of pixels is significantly brighter than other pixels of the target, which end up being ignored in the ASC extraction.

b) Extraction efficiency between real and synthetic data: In order to study the differences in extraction between synthetic and real data, we display Fig. 11 showing the ICC

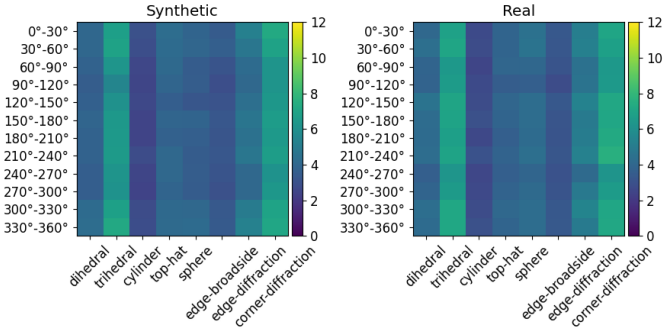


Fig. 8. Average number of reflectors for each radar effect.

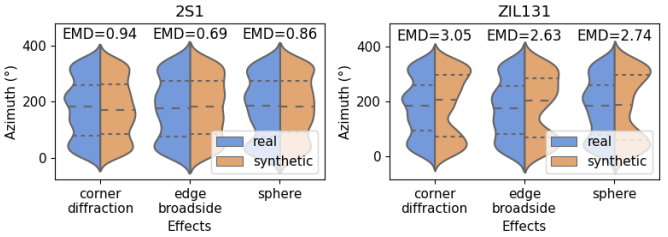


Fig. 9. Violin plot for the 2S1 and ZIL131 effects: synthetic vs real. For visualisation purposes, we only display three effects. The Earth Mover Distance (EMD) coefficient is also displayed for each effect. This calculates the difference between the real and synthetic distribution pairs. It can also be seen as the amount of change required for the synthetic distribution to equal the real one.

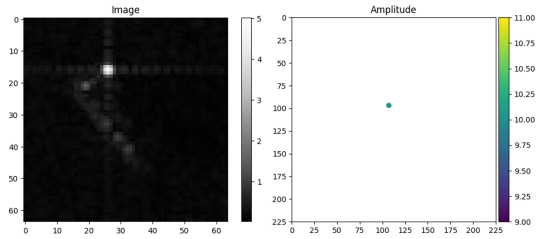


Fig. 10. Example of a pathologic case of the ZIL131 target, azimuth 125°. SAR Image is dispalyed on the left, while ASC parameters are shown on the right.

distribution via histograms for four different targets. Overall, better reconstructions are possible based on real data. This fact is somewhat reassuring, as it shows that ASC extraction reflects physical reality.

C. ASC data augmentation study for SAR ATR

Training configurations are completed as described in Table II. For all cases, the trained ATR is then tested on the same MSTAR real set and the results are provided in Table III with accuracy and F1-score.

Training the ATR model on reconstructed ASC dataset set results (Baseline ASC) in an accuracy of 55.88%, while adding ASC augmentations process (TSM9) ends up with 60.86%. Adding ASC augmentation process confers an improvement almost identical to adding image augmentation, which gives

TABLE III
EXPERIMENTAL RESULTS

Test name	Baseline	Augmented Baseline	Baseline ASC	TSM1	TSM2	TSM3	TSM4	TSM5	TSM6	TSM7	TSM8	TSM9	**TSM10**
Test Accuracy	53.25	59.58	55.88	54.63	58.8	51.67	62.21	54.3	60.81	56.62	52.78	60.86	**70.97**
F1-Score	51.24	58.5	55.39	53.45	57.79	49.93	60.24	53.07	59.79	54.88	51.49	59.83	**70.16**

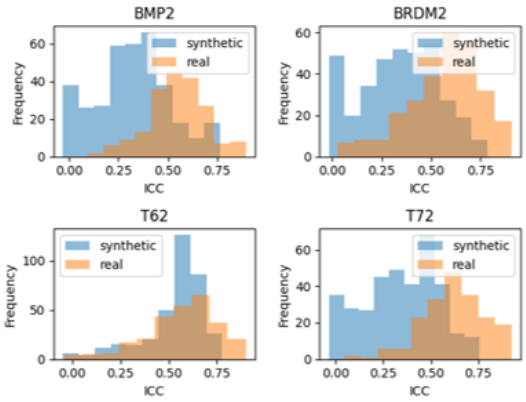


Fig. 11. ICC distribution: synthetic vs real (50 reflectors, 200 iterations).

a 59.58% accuracy. However, combining the two methods (TSM10) gives an accuracy of 70.97%.

According to the TSM4 test, adding new reflectors near the target seems to have a positive effect on performance. TSM6 and TSM9 tests confirm this trend, as they are also based on reflectors addition augmentations. This result could be explained by the fact that synthetic and measured samples may exhibit different target signatures, with scatterers amplitude and position slightly shifted.

We present our method as a complement to existing data augmentation techniques and compare it to those described in the literature. For example, Baffour et al. [1] use traditional augmentation and report 92% accuracy on MSTAR, but their results rely on the flawed SAMPLE dataset [2]. The approach by Camus et al. [2] is most comparable to ours: with their augmentation pipeline and MOCEM dataset, they reach 75% accuracy. The observed differences stem from the use of broader augmentation (affecting both target and clutter, versus only the target in our ASC-based DA), and more techniques employed (bagging, adversarial training, Test-Time Domain Augmentation). In order to quantify the isolated contribution of the data augmentation method, we can compare our TSM9, which achieves 60.86% accuracy, with their test incorporating the ResNet architecture and domain randomisation (both tests also involve random shifting), which achieves 50.48% accuracy. This result highlights the promise and robustness of our approach, which is independent of MOCEM and induces more localized changes.

V. CONCLUSION AND PERSPECTIVES

In this paper, we propose a data augmentation technique based on physical knowledge about SAR images to bridge the gap between synthetic and measured data. First, we presented the Attributed Scattering Centres Model, which is a physical model aiming at accurately describing the target information in SAR images. Next, we described our data augmentation method using this physical knowledge. In fact, having a set of ASC parameters, it is next possible to generate new samples by slightly perturbing them. These augmented samples can then be used to train models. Given the MSTAR and the corresponding synthetic datasets, we analyse different augmentation configurations and their impact on recognition performance. We showed that our proposed data augmentation pipeline can help gain up to 15.09% in terms of accuracy (see TSM10 vs Baseline ASC in Table III). Moreover, we identified that adding scattering mechanisms near the target is the most prolific augmentation, being able to give physical clues about the SAR S2M domain gap.

As a consequence, we believe that this approach shows potential for the future. Forthcoming work could involve the several following aspects.

Our ASC parameter estimation algorithm is not yet fully operational as it can be seen in Fig. 6. Some samples are not correctly reconstructed, leading to incomplete target information for some classes and azimuth angles, and thus deteriorating the representativeness of the dataset. As a consequence, we believe that our data augmentation pipeline and its corresponding results are not at the maximum of their potential. These moderate results can be explained by different reasons, for example by our way of initialising parameters. On the other hand, we need to further investigate and characterise the S2M domain gap. In fact, this paper shows some preliminary results that enabled us to test our data augmentation process. However, as explained earlier, we truly believe that we can also unveil which data augmentation method is most useful in the current task and why with our DA process. For example, we would like to be able to provide explanations such as: "we identified that for these measured and synthetic datasets, the rear of the 2S1 target is different because it shows dihedral instead of trihedral effects. As a consequence, we can guarantee domain generalisation by using the corresponding data augmentation during training." In other words, we would like to both characterise physically why synthetic and measured samples are different, and bridge the domain by using the most useful and justified data augmentation technique.

ACKNOWLEDGEMENT

We deeply thank the French MoD (DGA) for providing the synthetic dataset and reviewing this article. We also address our gratitude to SCALIAN DS for our constructive discussions.

REFERENCES

[1] A. A. Baffour, I. Osei Agyemang, I. Adjei-Mensah and R. E. Nuhoho, "Towards Fully Synthetic Training: Exploring Data Augmentations for Synthetic-to-Measured SAR in Automatic Target Recognition", 2025, IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/11105313,

[2] B. Camus, C. Le Barbu, E. Monteux, "Robust SAR ATR on MSTAR with Deep Learning Models trained on Full Synthetic MOCEM data", 2022, arXiv:2206.07352. [Online]. Available: https://arxiv.org/abs/2206.07352

[3] Y. Zhong, W. Zhou, and Z. Wang, 'A Survey of Data Augmentation in Domain Generalization', Neural Process Lett, vol. 57, no. 2, p. 34, Mar. 2025, doi: 10.1007/s11063-025-11747-9.

[4] M. Scherreik and K. Grubaugh, 'Linear separability of the SAR domain gap', in Algorithms for Synthetic Aperture Radar Imagery XXXII, E. Zelnio and F. D. Garber, Eds., Orlando, United States: SPIE, May 2025, p. 32. doi: 10.1117/12.3067969.

[5] N. Inkawhich et al., 'Bridging a Gap in SAR-ATR: Training on Fully Synthetic and Testing on Measured Data', IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing, vol. 14, pp. 2942–2955, 2021, doi: 10.1109/JSTARS.2021.3059991.

[6] A. W. Denton, 'DATA AUGMENTATION FOR SYNTHETIC APERTURE RADAR USING ALPHA BLENDING AND DEEP LAYER TRAINING'.

[7] J. Lv and Y. Liu, 'Data Augmentation Based on Attributed Scattering Centers to Train Robust CNN for SAR ATR', IEEE Access, vol. 7, pp. 25459–25473, 2019, doi: 10.1109/ACCESS.2019.2900522.

[8] B. Ding, G. Wen, X. Huang, C. Ma, and X. Yang, 'Data Augmentation by Multilevel Reconstruction Using Attributed Scattering Center for SAR Target Recognition', IEEE Geosci. Remote Sensing Lett., vol. 14, no. 6, pp. 979–983, Jun. 2017, doi: 10.1109/LGRS.2017.2692386.

[9] M. J. Gerry, "Two-dimensional inverse scattering based on the GTD model /," phdthesis, The Ohio State University, 1997. [Online]. Available: http://rave.ohiolink.edu/etdc/view?acc_num=osu1487946103567201.

[10] MSTAR database. Available online: https://www.sdms.afrl.af.mil/index.php?collection=mstar.

[11] C. COCHIN, P. POULIGUEN, B. DELAHAYE, D. l. HELLARD, P. GOSSELIN and F. AUBINEAU, "MOCEM - An 'all in one' tool to simulate SAR image," 7th European Conference on Synthetic Aperture Radar, Friedrichshafen, Germany, 2008, pp. 1-4. keywords: Solid modeling;Radar imaging;Synthetic aperture radar;Radar cross section;Computational modeling;Image resolution,

[12] B. Camus, S. Ds, J.-C. Louvigné, and C. Saleun, "Génération massive d'images SAR synthétiques pour l'IA à fins de classification automatique (ATR)," presented at ENVIREM 2025, Palaiseau, France, 2025.

Evolution of BA-LR and application to explainable cross-domain speaker verification*

Raphaël Duroselle
Inria Défense & Sécurité
LR2
France
raphael.duroselle@inria.fr

Yosra Jelassi
Inria Défense & Sécurité
LR2
France
yosra.jelassi@inria.fr

Jean-François Bonastre
Inria Défense & Sécurité
LR2
France
jean-francois.bonastre@inria.fr

***Abstract*—The Binary-Attribute Likelihood-Ratio (BA-LR) method has been proposed as an explainable speaker verification system focusing on forensic applications. BA-LR represents a speech utterance by a binary vector, indicating the presence or absence of speech attributes. In this work, we introduce a better founded formulation of BA-LR that can handle more naturally enrollments with multiple recordings. In addition this new formulation allows incorporating into the weight of evidence of each attribute its robustness to mismatch between enrollment and test conditions, leading to cross-domain scoring.**

***Index Terms*—speaker recognition, explainability, BA-LR, domain adaptation, NIST SRE24**

I. Introduction

The speaker verification task (SV) consists in deciding whether one test utterance was pronounced by a given speaker, represented by one or more enrollment recordings. Variability between enrollment and test conditions is a key factor that can limit system performance. This variability also adds a potential level of uncertainty to system reliability. When both conditions are known, this variability is denoted "domain mismatch" or "cross domain condition". For instance, recent NIST SRE campaigns [1] have focused on two challenging cross-domain tasks: cross-language and cross-source speaker verification.

Numerous methods have been proposed to model variability between conditions and improve performance accordingly [2]–[6]. Between them, the 4-cov PLDA [7] introduces an explicit model of the dependence between speaker embedding distributions over two domains and can be used for cross-domain scoring. These approaches are complementary to normalizations of the embeddings to limit variability between genders, languages or channels [8].

Recent SV systems [9]–[11] are based on high-dimensional embeddings, similar to x-vectors [12]. These systems use a large neural model with millions of parameters trained on large, poorly controlled databases. Thanks to their ability to exploit this considerable amount of data and parameters, they are able to discriminate between speakers and manage session variability, thus achieving cutting-edge performance. However, these systems produce a single score per trial and are unable to link this score, or parts of it, to a subset of input features, certain training examples, or certain parts of the model. They also often show a significant loss of performance when a domain mismatch occurs, i.e., when real-life conditions differ from training conditions, and this loss is difficult to predict. These two aspects lead to a lack of explainability and reliability that can significantly limit the deployment of practical solutions based on these systems. Furthermore, explainability itself is becoming increasingly necessary due to regulations such as the EU's GDPR or AI laws, and is mandatory in areas such as forensic and investigative speaker recognition [13]. Explainability is also necessary for reliability, because understanding and describing how a system works is essential for certifying its performance under specific working conditions.

Among several publications on explanability in speaker recognition [14]–[16], the BA-LR method [17], [18] has some interesting and specific features. First, BA-LR is an intrinsically explainable approach that offers three levels of explainability/interpretability: modelling, scoring and drivers. BA-LR models a speech utterance using a binary vector (BA) where a coefficient explicitely indicates the presence or absence of a given speech attribute. Each of the several hundreds attributes is shared among a group of speakers. The BA extractor is trained similarly to recent speaker embedding extractors, without requiring additional labels. It takes full advantage of x-vector state-of-the-art approaches but differs in that it produces binary embeddings and in the behavior of a coefficient: here, $BA(i) = 1$ means that the utterance contains the attribute i. The second level of explainability is scoring. BA-LR generates log-likelihood ratios (LLR) for each attribute (BA coefficient) and the final LLR is a simple combination of them. Finally, the third level is an explanation of the attributes, providing an analysis of the underlying phonetic factors. This explainability phase is done after training the system [19]. BA-LR also differs from the main part of other approaches in that it was introduced for forensic applications and has been evaluated in this context using a realistic database [20]. Recently, it was also applied to explainable spoofed speech characterization [21].

In this work, we do not address the tasks of extracting [18] or characterizing binary attributes [19] but focus on formulating the BA-LR speaker verification scoring based on statistics

This work has been partly funded by the Convention IA DGA - project VoxProfil.

of activation of binary attributes, with the aim of improving performance in cross-domain trials. Indeed, standard feature-based adaptation methods [22] are not meant to preserve the binary structure of embeddings, while model-based cross-scoring methods do not meet the explainability standards of BA-LR [5], [7]. Consequently we propose a new formulation of the BA-LR approach, which takes domain modeling into account.

In this paper, we make several contributions to the BA-LR method for speaker verification. We introduce BA-LR-v2, a new probabilistic formulation of the BA-LR model, and implement it with a Beta-Bernoulli model [23], [24]. We propose a simple cross-domain extension of BA-LR scoring and implement it with a Gaussian copula. This new BA-LR-v2 with cross-domain modeling is experimentally validated, both on a controlled experiment on VoxCeleb with simulated channel degradation and on the challenging NIST SRE24 corpus with cross-source trials. It is particularly effective at handling multiple enrollment utterances and cross-domain trials and incorporates the sensitivity of each attribute to domain mismatch into the weight of evidence.

II. BA-LR-v2 scoring

The BA-LR scoring model can be summarized by three hypotheses.

1) The test and enrollment sets of utterances x_t and x_e are represented as counts of activations a_i and non-activations n_i of N binary attributes.
$$x_t = (a_1^t, n_1^t \dots a_N^t, n_N^t) \quad x_e = (a_1^e, n_1^e \dots a_N^e, n_N^e) \tag{1}$$
2) The binary attributes are independent, which implies that the LLR can be decomposed into contributions LLR_i from each attribute. This assumption has been partially verified in [18].
$$LLR(x_t, x_e) = \sum_{i=1}^{N} LLR_i[(a_i^t, n_i^t), (a_i^e, n_i^e)] \tag{2}$$
3) For a given attribute i, LLR_i depends on statistics of activation of the attribute among a reference population.

In this work, we propose a new formulation of the LLR. We call it BA-LR-v2. We refer to [25] for an in-depth description of the original BA-LR model. In the two following sections, we work with a single binary attribute and omit the attribute index i.

A. Proposed formulation: BA-LR-v2 scoring

We assume that a speaker is represented by a latent variable p corresponding to the frequency of activation of the attribute in an utterance. The activation of the attribute for this speaker follows a Bernoulli distribution with parameter p.

We define the distribution of this probability of activation among a reference population of speakers and call its density f. According to this model, the likelihood of a sequence of observations of the attribute for a given speaker, with counts of activations (a, n), is given by:

$$L(a, n) = \int_{p=0}^{1} p^a (1-p)^n f(p) dp \tag{3}$$

The speaker verification LLR is obtained by grouping differently counts of activation of enrollment and test observations of the attribute, in a similar way to [7] for PLDA.

$$LLR = \log \frac{L(a_t + a_t, n_e + n_t)}{L(a_e, n_e) L(a_t, n_t)} \tag{4}$$

The spirit of the original BA-LR model can be found by selecting a distribution f with only two possible values of the probability of activation, interpreted as the groups of speakers with and without the attribute.

B. Implementation of BA-LR-v2 with a Beta-Bernoulli model

Similar to [23], [24], choosing a Beta distribution, with parameters α and β, simplifies the computation of the likelihood, since it is the conjugate of the Bernoulli distribution.

$$f(p|\alpha, \beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)} \tag{5}$$

where $B(\alpha, \beta) = \int_{p=0}^{1} p^{\alpha-1}(1-p)^{\beta-1} dp$ is the Beta function.

$$L(a, n) = \frac{B(\alpha + a, \beta + n)}{B(\alpha, \beta)} \tag{6}$$

Consequently the LLR is given by:

$$LLR = \log \frac{B(\alpha + a_t + a_e, \beta + n_t + n_e) B(\alpha, \beta)}{B(\alpha + a_t, \beta + n_t) B(\alpha + a_e, \beta + n_e)} \tag{7}$$

III. Cross-domain scoring with BA-LR-v2

The distribution of attribute activation varies depending on the domain. This may be due to a domain mismatch with the training corpus of the attribute extractor, to noise levels that erase the attribute in an utterance, or to attributes that disappear under some conditions. For example, the fundamental frequency is outside of the telephone bandwidth for most speakers [26].

A. Cross-domain model

We propose to model the variability between probabilities of activation of an attribute under different conditions. We note 1 and 2 the two domains. A sequence of utterances from the same speaker is represented by counts of activation and non activation of the attribute for each condition.

$$x = \begin{pmatrix} a^1 & n^1 \\ a^2 & n^2 \end{pmatrix} \tag{8}$$

We now assume that each speaker is characterized by the probabilities p_1 and p_2 of activating the attribute on each condition. We assume a relationship between these two latent variables similar to the 4-cov PLDA model [7] and denote $f(p_1, p_2)$ the joint density of these probabilities among a reference population of speakers. Then, the likelihood of a

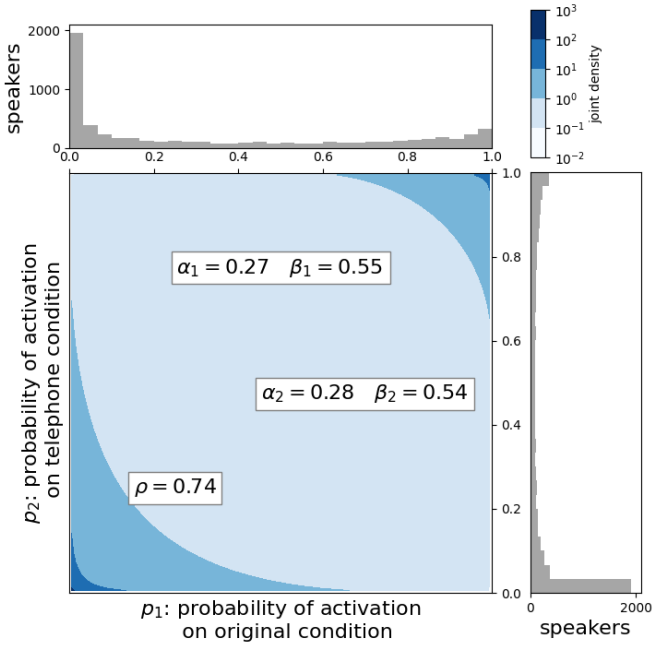


Fig. 1: Histogram of per-speaker probability of activation on each domain and estimated joint density for attribute BA386 (VoxCeleb protocol).

LLR		x_t 0	x_t 1
x_e	0	0.24	-0.80
	1	-0.80	0.75

mono-domain: original

LLR		x_t 0	x_t 1
x_e	0	0.25	-0.80
	1	-0.80	0.72

mono-domain: telephone

LLR		x_t 0	x_t 1
x_e	0	0.17	-0.47
	1	-0.47	0.55

cross-domain: telephone/original

Fig. 2: LLR values with cross-domain BA-LR-v2 for attribute BA386 (VoxCeleb protocol).

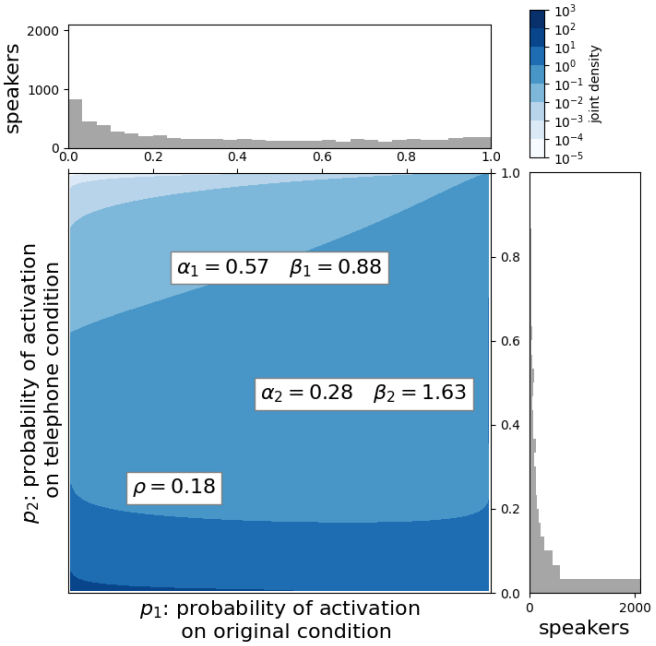


Fig. 3: Histogram of per-speaker probability of activation on each domain and estimated joint density for attribute BA220 (VoxCeleb protocol).

LLR		x_t 0	x_t 1
x_e	0	0.23	-0.52
	1	-0.52	0.49

mono-domain: original

LLR		x_t 0	x_t 1
x_e	0	0.06	-0.42
	1	-0.42	1.10

mono-domain: telephone

LLR		x_t 0	x_t 1
x_e	0	0.02	-0.03
	1	-0.12	0.16

cross-domain: telephone/original

Fig. 4: LLR values with cross-domain BA-LR-v2 for attribute BA220 (VoxCeleb protocol).

sequence of utterances belonging to the same speaker is given by:

$$L = \iint p_1^{a^1}(1-p_1)^{n^1} p_2^{a^2}(1-p_2)^{n^2} f(p_1,p_2)dp_1dp_2 \quad (9)$$

B. Implementation with copula

The likelihood can be computed directly, using the empirical joint distribution of a training population of speakers. In practice, it may be convenient to consider modeling of the marginal distributions of p_1 and p_2 separately from the dependence structure between the two variables. For example, the marginal distributions may be estimated on two large corpora representative of each condition whereas we need a corpus with observations of the same speakers on both conditions to estimate the dependence between the two variables. The dependence structure between the two variables can be modeled with a copula [27]. In our context it is a bivariate cumulative density function defined on $[0,1]^2$ with uniform marginal distributions. We model the joint density by a copula of density $c(u,v)$ and the marginal distributions with cumulative density functions F_1, F_2 and densities f_1, f_2.

$$f(p_1,p_2) = c[F_1(p_1),F_2(p_2)]f_1(p_1)f_2(p_2) \quad (10)$$

In this first implementation, we retain Beta distributions for f_1 and f_2 and select a Gaussian copula [28] for c. It has a single parameter $\rho \in]-1,1[$ which encodes the correlation between the two distributions. ρ can be estimated by maximum likelihood on a set of utterances with the same speakers on both conditions. A better model could be selected by an analysis of the actual joint distributions.

$$c(p_1',p_2'|\rho) = \frac{\mathcal{N}((\Phi^{-1}(p_1'),\Phi^{-1}(p_2'))|0,R)}{\mathcal{N}(\Phi^{-1}(p_1')|0,1)\mathcal{N}(\Phi^{-1}(p_2')|0,1)} \quad (11)$$

where Φ^{-1} is the inverse of the cumulative densitiy function of a standard normal distribution $\mathcal{N}(.|0,1)$, and $\mathcal{N}(.|0,R)$ is a multivariate normal distribution with covariance matrix R:

$$R = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \tag{12}$$

The Gaussian copula has been recently applied to speaker recognition for score-level system fusion [29]. In [29], it models the dependence between the distributions of scores of different systems, whereas in our work it models the dependence between the distributions of a latent variable on two domains.

With the cross-domain model, likelihood values are no longer in closed form and require more computation. In a practical scenario with a large numbers of trials, the maximum number of observations of the attribute is often known in advance (one utterance for each enrollment on VoxCeleb1, three utterances maximum on NIST SRE24) and all required likelihood values can be precomputed once before inference.

Figures 1, 2, 3, 4 illustrate the impact of the latent variable joint distribution model on scoring, for attribute BA386 (high correlation between the two domains) and attribute BA220 (low correlation). Histograms of activation probabilities per speaker on each domain are shown on the x and y axes, while the estimated joint density is shown in the center, along with the corresponding values of the α_1, β_1, α_2, β_2 and ρ parameters. The impact on LLR values is shown in Figures 2 and 4. A weak correlation between the domains results in lower absolute LLR values for cross-domain trials.

IV. Experiments

To assess the validity of the proposed BA-LR scoring, we conduct two sets of experiments, first on a simulated and controlled corpus and then on the challenging NIST SRE24 corpus. For both sets of experiments, we use a baseline speaker verification system based on 256-dimensional embeddings. We extract vectors of activation of 512 attributes from these embeddings with a binary autoencoder (BAE) trained with the protocol described in [18]. We then apply the original BA-LR (speech-oriented model in [20]), as well as the proposed BA-LR-v2 scoring method.

A. VoxCeleb protocol with simulated cross-domain trials

The goal of this first experiment is to check the validity of the proposed cross-domain scoring with a controlled mismatch between the two conditions. We train the systems on VoxCeleb2 [30] and evaluate them on VoxCeleb1 [31]. The first condition consists of the original VoxCeleb utterances, corresponding to audio from video, with a sampling rate of 16kHz. For the second condition we apply a telephone bandpass filter (300-3400Hz) to the original files. Cross-condition trials are constituted of an enrollment utterance from the simulated telephone condition and an original test utterance.

The baseline system is the Wespeaker ResNet34 model with large-margin finetuning, cosine similarity scoring, and without AS-norm [11]. The BAE and BA-LR parameters are estimated on VoxCeleb2.

B. NIST SRE24 protocol

The NIST SRE24 corpus [1] focuses on challenging conditions: cross-lingual trials, among Tunisian Arabic, French and English, and cross-source trials, including conversational telephone speech (CTS) and audio from video (AfV). We evaluate the proposed cross-domain scoring method on cross-source trials. The two conditions differ not only by bandwidth but also by the level of noise, the speech content, and even the number of speakers contained in the test utterance. In addition, the NIST SRE24 corpus contains trials with multiple enrollment utterances.

The baseline speaker verification system is a ResNet101 model developed with the kiwano toolkit[1]. It is trained on the CTS superset corpus [32] (only on telephone condition), and all data is downsampled to 8 kHz. The speaker verification system is trained with the Jeffreys loss [33]. At inference, non speech regions are removed with rVAD [34]. For evaluation of the baseline system, a simple preprocessing step is applied to the embeddings before cosine similarity scoring (centering, reduction to 100 dimensions with LDA, and length normalization), whereas the original 256-dimensional embeddings are used for the extraction of binary attributes. The BAE is trained on the CTS superset corpus. BA-LR parameters are estimated on the SRE21-eval audio corpus which contains both conditions (CTS and AfV) but different languages (Cantonese, English, and Mandarin). For all systems, a per-condition logistic regression is trained on the SRE24-dev corpus. We use six conditions corresponding to the columns of Table II and defined by the enrollment and test channels (CTS or AfV) and the number of enrollment utterances (1 or 3).

V. Results

A. VoxCeleb experiments

Evaluation of the systems is reported in terms of equal error rate (EER) in Table I. We report performance on three conditions corresponding to matched conditions with original or simulated telephone utterances, and to cross-domain scoring with telephone enrollment utterances and original test utterances[2].

The original Wespeaker model achieves competitive performance on the original VoxCeleb1 [11]. It suffers from a significant but limited performance drop on the unseen simulated telephone condition. This performance drop is more moderate for cross-domain trials (twice the error rate of *original*).

The other systems correspond to different scoring methods with the same binary attributes. Using binary attributes for scoring with cosine similarity produces a performance drop,

[1] https://github.com/mrouvier/kiwano

[2] For consistency with published results, we report performance with three digits. However, a calculation of 95 % confidence intervals using a bootstrap (1000 samplings) taking into account speaker labels [35] provides the following intervals for the baseline *ResNet34-LM* model on *original/original* condition: $[0.51, 1.13]$ on *O-clean*, $[0.88, 1.09]$ on *E-clean*, and $[1.63, 1.94]$ on *H-clean*.

TABLE I: Performance on VoxCeleb1 (best BA-LR performance in bold). BA refers to binary attributes.

	System	EER (%) on VoxCeleb1 *[enrollment condition]/[test condition]*								
		original/original			**telephone/telephone**			**telephone/original**		
		O-clean	E-clean	H-clean	O-clean	E-clean	H-clean	O-clean	E-clean	H-clean
	ResNet34-LM + cosine similarity [11]	0.814	0.933	1.695	1.803	2.123	4.031	1.462	1.881	3.488
	cosine similarity	1.212	1.322	2.278	2.978	3.293	5.815	2.579	3.198	5.155
BA	BA-LR *original*	1.255	1.443	**2.418**	3.286	3.645	6.247	2.973	3.662	5.677
BA	BA-LR-v2 *original*	**1.234**	**1.343**	2.445	2.755	3.007	5.674	2.345	2.792	5.027
BA	BA-LR-v2 *telephone*	1.404	1.523	2.756	**2.462**	**2.738**	**5.101**	2.489	2.962	5.202
BA	cross-domain BA-LR-v2	-	-	-	-	-	-	**2.287**	**2.613**	**4.879**

TABLE II: Performance on NIST SRE24-eval audio (best BA-LR performance in bold). BA refers to binary attributes.

	System	SRE24 eval audio			EER (%) on subset of trials *[enrollment condition]-[# enrollment utterances]/[test condition]*					
		$C_{Primary}$ *min*	$C_{Primary}$ *act*	EER (%)	CTS-1/CTS	CTS-3/CTS	AfV-1/AfV	AfV-1/CTS	CTS-1/AfV	CTS-3/AfV
	ResNet101 + cosine similarity	0.698	0.830	10.47	4.73	2.48	7.29	8.06	8.77	6.85
	cosine similarity	0.778	0.861	12.90	5.65	2.93	8.77	10.74	11.41	9.01
BA	BA-LR	0.794	**0.801**	12.90	5.37	4.43	8.44	10.18	10.74	10.21
BA	BA-LR-v2	**0.768**	0.812	12.62	**5.32**	**2.69**	8.55	9.96	10.52	7.83
BA	cross-domain BA-LR-v2	0.782	0.818	**12.46**	5.58	2.81	**8.33**	**9.57**	**10.15**	**7.68**

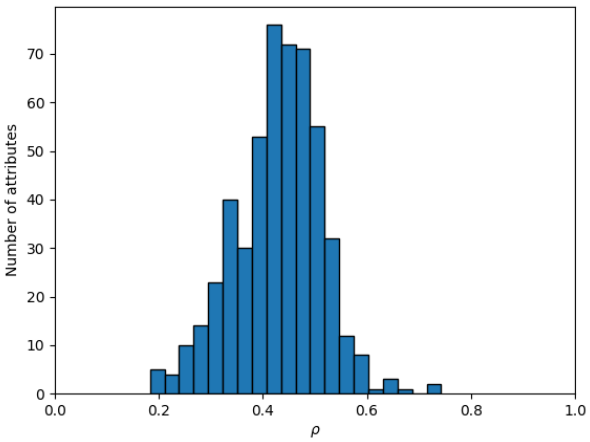


Fig. 5: Histogram of values of the Gaussian copula correlation parameter (VoxCeleb protocol) for the 512 attributes. The parameter models the dependency between probabilities of activation of an attribute on each domain.

similar on matched and unmatched conditions, for instance from 4.031% to 5.815% on H-clean $telephone/telephone$. The BA-LR model trained on $original$ data achieves worse performance than cosine similarity, especially for telephone and cross-domain conditions. The proposed BA-LR-v2 trained on $original$ data outperforms cosine similarity on the unknown telephone condition and on the cross-domain condition. BA-LR-v2 achieves its best performance when its parameters are trained on matched conditions (BA-LR-v2 $original$ and $telephone$).

Finally, the proposed cross-domain BA-LR-v2 method achieves the best performance for cross-domain trials. This improvement is statistically significant only on the *E-clean* trial list. Figure 5 represents the distribution of the estimated value of the correlation parameter ρ. Most of the attributes exhibit a weak to moderate correlation (between 0.2 and 0.6). The cross-domain BA-LR-v2 model weights the contribution of each attribute according to this correlation parameter, as illustrated by Figures 1, 2, 3, 4 for attributes BA386 and BA220.

B. NIST SRE24 experiments

Evaluation of the systems trained on the NIST SRE corpus is reported in Table II. The evaluation corpus is SRE24-eval audio, and we report the official $C_{Primary}$, $minC_{Primary}$ and EER [1]. In addition, we report the EER on specific subsets defined by the enrollment and test channels (CTS or AfV) and the number of enrollment utterances (1 or 3).

The baseline ResNet101 system achieves a $C_{Primary}$ of 0.830 on the challenging SRE24-audio-eval corpus. The binarization of the embeddings produces an important drop in performance, more pronounced in terms of EER than in terms of $C_{Primary}$. BA-LR scoring trained on a corpus containing both CTS and AfV matches cosine similarity for enrollment with a single utterance. The proposed BA-LR-v2 achieves the same overall performance as BA-LR, but with a strong improvement for trials with multiple enrollment utterances. The cross-domain BA-LR-v2 improves discrimination performance for the cross-source trials. Overall, BA-LR systems match the performance of the baseline ResNet101 system in terms of $C_{Primary}$, corresponding to low false alarm operating points, but not in terms of EER.

VI. Discussion

BA-LR is evaluated for the first time on the challenging NIST SRE campaign, demonstrating that explainable systems can be built with a moderate degradation of performance. We reach or improve the performance of cosine similarity scoring on binary attributes with BA-LR scoring, demonstrating that the drop in performance is not due to the explainable scoring mechanism but only to the binary attribute extraction step. Improving this process, particularly to ensure independence of the attributes, could help reduce the performance gap with the baseline system.

The new BA-LR-v2 scoring achieves better performance than the original BA-LR formulation for trials with multiple enrollment utterances. In addition, it leads to a very natural cross-domain scoring that relies on the modeling of the dependence between attribute activations on both conditions. From an explainability point of view, the modeling of the dependence between probabilities of activations on two conditions is crucial because it allows the weight of evidence of each attribute to be balanced with the robustness of the attribute across conditions.

Our first implementation with Gaussian copula leads to a limited performance improvement for cross-domain scoring. A better choice of the model of dependence between the conditions could make this method more efficient, for instance exploring other families of copulae.

VII. Conclusion

We introduce BA-LR-v2, a new formulation of BA-LR, that bridges the gap with classical speaker verification scoring models and improves performance for trials with multiple enrollment utterances. In addition, we model the dependence between the frequencies of activation of an attribute on two conditions, which enables cross-domain speaker verification scoring. Our implementation with a Beta-Bernoulli model and a Gaussian copula gives consistent improvements over the original BA-LR model, both in a controlled experiment on VoxCeleb1 with a simulated channel degradation and on the challenging NIST SRE24 corpus with cross-source trials. This reduces the performance gap between the explainable BA-LR system and state-of-the-art speaker verification systems, while bringing a new dimension of explainability by introducing into the weight of evidence of each attribute its robustness to condition mismatch.

Acknowledgment

This work was performed using HPC resources from GENCI–IDRIS (Grant 2025-AD011014982R1). We thank: Imen Ben-Amor for introducing the BA-LR model, Mickaël Rouvier for training the ResNet101 model, Lukáš Burget and Anna Silnova for the idea of Beta-Bernoulli scoring for BA-LR and Thibaut Vasseur for advice about copula.

References

[1] NIST. , "NIST 2024 Speaker Recognition Evaluation Plan," 2024.

[2] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1, pp. 42–54, Jan. 2000.

[3] D. Reynolds, "Channel robust speaker verification via feature mapping," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 2, Apr. 2003, pp. II–53.

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[5] L. Ferrer, M. McLaren, and N. Brummer, "A speaker verification backend with robust performance across conditions," *Computer Speech & Language*, vol. 71, p. 101258, Jan. 2022.

[6] H. Zhang, L. Wang, K. A. Lee, M. Liu, J. Dang, and H. Chen, "Meta-learning for cross-channel speaker verification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 5839–5843.

[7] P.-M. Bousquet and M. Rouvier, "Duration mismatch compensation using four-covariance model and deep neural network for speaker verification," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 1547–1551.

[8] J. Alam, S. Barahona, D. Boboš, L. Burget, S. Cumani, M. Dahmane, J. Han, M. Hlavaček, M. Kodovsky, F. Landini, L. Mošner, P. Palka, T. Pavliček, J. Peng, O. Plchot, G. P. Rajasekhar, J. Rohdin, A. Silnova, T. Stafylakis, and L. Zhang, "ABC system description for NIST SRE 2024."

[9] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, Aug. 2021.

[10] J. Huh, J. S. Chung, A. Nagrani, A. Brown, J.-W. Jung, D. Garcia-Romero, and A. Zisserman, "The VoxCeleb speaker recognition challenge: A retrospective," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3850–3866, 2024.

[11] S. Wang, Z. Chen, B. Han, H. Wang, C. Liang, B. Zhang, X. Xiang, W. Ding, J. Rohdin, A. Silnova, Y. Qian, and H. Li, "Advancing speaker embedding learning: Wespeaker toolkit for research and production," *Speech Communication*, vol. 162, p. 103104, Jul. 2024.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5329–5333.

[13] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J.-F. Bonastre, and D. Matrouf, "Forensic speaker recognition," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 95–103, Mar. 2009.

[14] T. Thebaud, G. Hernandez Sierra, S. F. Samson Juan, and M. Tahon, "A Phonetic Analysis of Speaker Verification Systems through Phoneme selection and Integrated Gradients," in *Speaker and Language Recognition Workshop - Odyssey*, Quebec, Canada, Jun. 2024. [Online]. Available: https://hal.science/hal-04578447

[15] Y. Ma, S. Wang, T. Liu, and H. Li, "Expo: Explainable phonetic trait-oriented network for speaker verification," *IEEE Signal Processing Letters*, 2025.

[16] X. Liu, J. Yamagishi, M. Sahidullah, and T. Kinnunen, "Explaining speaker and spoof embeddings via probing," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[17] I. Ben-Amor and J.-F. Bonastre, "BA-LR: Binary-Attribute-based Likelihood Ratio estimation for forensic voice comparison," in *2022 International Workshop on Biometrics and Forensics (IWBF)*. Salzburg, Austria: IEEE, Apr. 2022, pp. 1–6.

[18] I. Ben-Amor, J.-F. Bonastre, and S. Mdhaffar, "Extraction of interpretable and shared speaker-specific speech attributes through binary auto-encoder," in *Interspeech 2024*. ISCA, Sep. 2024, pp. 3230–3234.

[19] I. Ben-Amor, J.-F. Bonastre, B. O'Brien, and P.-M. Bousquet, "Describing the phonetics in the underlying speech attributes for deep and interpretable speaker recognition," in *Interspeech 2023*. ISCA, Aug. 2023, pp. 3207–3211.

[20] I. Ben-Amor, J.-F. Bonastre, and D. V. D. Vloed, "Forensic speaker recognition with BA-LR: Calibration and evaluation on a forensically realistic database," in *The Speaker and Language Recognition Workshop (Odyssey 2024)*. ISCA, Jun. 2024, pp. 9–16.

[21] J. Mishra, M. Chhibber, H.-j. Shim, and T. H. Kinnunen, "Towards explainable spoofed speech attribution and detection: A probabilistic approach for characterizing speech synthesizer components," *Submitted to Computer Speech and Language*, Feb. 2025.

[22] P.-M. Bousquet and M. Rouvier, "Adaptation strategy and clustering from scratch for new domains of speaker recognition," in *The Speaker and Language Recognition Workshop (Odyssey 2020)*. ISCA, Nov. 2020, pp. 81–87.
[23] J. Alam, P. Kenny, G. Bhattacharya, and M. Kockmann, "Speaker verification under adverse conditions using i-vector adaptation and neural networks," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 3732–3736.
[24] O. Plchot, P. Matějka, A. Silnova, O. Novotný, M. D. Sánchez, J. Rohdin, O. Glembek, N. Brümmer, A. Swart, J. Jorrín-Prieto, P. García, L. Buera, P. Kenny, J. Alam, and G. Bhattacharya, "Analysis and Description of ABC Submission to NIST SRE 2016," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 1348–1352.
[25] I. Ben-Amor, "Deep modeling based on voice attributes for explainable speaker recognition: Application in the forensic domain," Ph.D. dissertation, 2024.
[26] N. Gleiss, "The effect of bandwidth restriction on speech transmission quality in telephony." in *Proc. 4th Int. Symp. on Human Factors in Telephony (Bad Wiessee, 1968)*. VDE-Verlag, 1970, pp. 1–6.
[27] R. B. Nelsen, *An Introduction to Copulas*, 2nd ed., ser. Springer Series in Statistics. New York: Springer, 2006.
[28] C. Meyer, "The Bivariate Normal Copula," *Communications in Statistics - Theory and Methods*, vol. 42, no. 13, pp. 2402–2422, Jul. 2013.
[29] S. Cumani, "A copula-based generative score-level fusion model for speaker verification," in *Interspeech 2025*, pp. 3723–3727.
[30] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Interspeech 2018*. ISCA, Sep. 2018, pp. 1086–1090.
[31] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 2616–2620.
[32] O. Sadjadi, "NIST SRE CTS Superset: A large-scale dataset for telephony speaker recognition," in *NIST*, 2021.
[33] P.-M. Bousquet and M. Rouvier, "Jeffreys Divergence-Based Regularization of Neural Network Output Distribution Applied to Speaker Recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.
[34] Z.-H. Tan, A. kr. Sarkar, and N. Dehak, "rVAD: An unsupervised segment-based robust voice activity detection method," *Computer Speech & Language*, vol. 59, pp. 1–21, Jan. 2020.
[35] L. Ferrer and P. Riera, "Confidence Intervals for evaluation in machine learning," https://github.com/luferrer/ConfidenceIntervals.

Hybrid Canonicalization of Intelligence Knowledge Graphs: A Frugal and Explainable Approach

Pauline Degez
ChapsVision
pdegez@chapsvision.com

Louis Jourdain
ChapsVision
ljourdain@chapsvision.com

***Abstract*—We present a frugal and explainable pipeline for soft canonicalization in large, noisy Knowledge Graphs, designed for intelligence applications. Unlike traditional clustering-based approaches, our method combines blocking with string similarity, a lightweight supervised classifier leveraging linguistic and graph features, and selective Large Language Model (LLM) validation at the decision frontier. Applied to a 180k-document corpus on the Ukraine conflict, the pipeline achieves high precision (0.998) while reducing LLM calls by up to 5× compared to LLM-only baselines. Results demonstrate scalability to millions of nodes, strong precision–recall trade-offs, and interpretable outputs, enabling more trustworthy knowledge fusion in high-stakes contexts.**

***Index Terms*—Canonicalization, Knowledge graphs, deduplication, Hybrid NLP pipeline, Large Language Models (LLMs), Intelligence analysis, Information extraction.**

I. Introduction

When processing vast volumes of textual data, Knowledge Graphs (KGs), widely used in downstream applications such as search [1], question answering [2], and recommendation, offer an efficient way to build structured representations of the information contained within a database. These capabilities are particularly valuable in intelligence and investigative scenarios, where it is critical to retrieve all pieces of information connected to a specific actor, event, or network [3]. KG utility is often limited by construction noise. Ideally, each node maps to one real-world entity, but NLP extraction from unstructured text yields multiple surface forms for the same entity. Variation stems from synonymy, typos/orthography/transliteration, and preprocessing errors. In high-stakes settings, such duplication distorts analysis, obscures links, and wastes resources. Hence, pipelines need a deduplication/canonicalization layer. Yet robust, scalable canonicalization for large, noisy KGs remains an open challenge. While deduplication has been widely studied for structured data, no flexible, effective, and scalable solution exists to unify entity mentions in large, heterogeneous real-world corpora. Many approaches depend on heavy supervised training or features unavailable in operational intelligence contexts [4]. Large Language Models (LLMs) offer strong language understanding, yet applying them directly to multi-million-node KGs is prohibitively costly; discarding them altogether would, however, forgo valuable capabilities [5]. This motivates an architecture that is (i) easily deployable across diverse technical stacks, (ii) independent of extensive training or rare features, (iii) able to integrate heterogeneous signals (textual, graph-based, and contextual)through rule-based NLP, similarity measures, machine learning, and selective LLM use, and (iv) scalable to Big Data intelligence scenarios. A further limitation in current research is the mismatch between evaluation datasets and real-world challenges. Common benchmarks such as ReVerb45K [4] or COMBO [6] are small, lack realistic noise, and contain few challenging minimal pairs, cases where two entities have highly similar surface forms and contexts but refer to different real-world entities. Moreover, these benchmarks rarely evaluate precision–recall trade-offs, despite the fact that in high-stakes applications false merges (incorrectly merging distinct entities) are often far more damaging than missed merges.

We therefore focus on a task we call *soft canonicalization*, a precision-first approach to entity normalization in Knowledge Graphs. Rather than exhaustively clustering all mentions into canonical entities, soft canonicalization only merges those cases where all occurrences can be confidently attributed to the same real-world entity. Ambiguous mentions are deliberately left unmerged, reducing the risk of false merges that could distort analysis in high-stakes scenarios.

In this work, we present a hybrid, cost-effective architecture for trustworthy and explainable KG canonicalization, designed for large-scale, noisy, intelligence-oriented datasets. We evaluate it on a domain-specific corpus covering the ongoing conflict in Ukraine to challenge its scalability, accuracy, and cost savings compared to purely LLM-driven approaches.

The main contributions of this work are:

- Highlighting the limitations of state-of-the-art KG canonicalization in intelligence contexts and introducing *soft canonicalization* as a precision-first alternative for high-stakes datasets;
- Presenting a hybrid architecture that combines classical similarity metrics, form- and graph-based heuristics, and selective LLM invocation to optimize precision–recall trade-offs;
- Reducing LLM cost through targeted, uncertainty-based usage, applying LLMs only at the decision frontier;
- Ensuring interpretability so that analysts can trace and trust the reasoning behind each merge.

II. Background and Related Work

A. On Knowledge Graphs and Knowledge Bases

Working with large unstructured corpora is difficult: even embeddings or vector search often lack the precision and contextual depth needed for reliable navigation. Downstream tasks are more effective when data is converted into structured representations. Among the most popular options, KG represents knowledge as triples such as (Obama, was born in, Honolulu), where nodes are entities or mentions and edges are relations. KGs differ in schema complexity, ranging from simple untyped co-occurrences to typed relations and entity-typed nodes. Designing them always involves a trade-off: stricter schemas yield precision and consistency, while looser schemas facilitate coverage and easier extension.

KG construction ties to Information Extraction (IE). In Closed IE, relations come from a predefined schema, but coverage is limited [7]. In Open IE, triples are extracted without schema constraints, with relations often realized as verbal expressions [8]. For investigative contexts with vast, heterogeneous, noisy sources, Open KGs are especially relevant [9]. However, Open IE extracts mentions, not canonical entities: aggregating mentions builds connections, yet synonyms and variants remain fragmented and queries may miss relevant results or conflate distinct real-world entities. Thus, KG building targets: (1) coverage (capture entities and relations), (2) precision (correct segmentation and labeling), and (3) entity canonicalization (merge mentions referring to the same entity). This last step is a full NLP task and is essential to maximize utility; a graph where duplicates are merged is sometimes called an Entity-Resolved Knowledge Graph.

B. Defining Deduplication

The presence of duplicates in databases has become increasingly prominent with the rise of Big Data [10] .The concept of deduplication spans multiple NLP tasks, defined by the nature of the objects being merged and the resources available to do so. Literature identifies a variety of related terms : author-name disambiguation, coreference resolution, entity linking, identity resolution, object consolidation, record linkage, schema matching [11]. Deduplication, in its broadest sense, refers to merging identical objects that appear multiple times in a dataset. In structured databases, it is achieved through field-level pattern matching or rules which is not applicable to Open KGs. Entity Linking connects textual mentions to entries in an existing knowledge base (KB) such as Wikidata. While this enforces the principle that each KG node corresponds to a single real-world entity, the approach depends entirely on external resources that contain errors, are incomplete, and rarely cover lesser-known individuals or organizations which is a major limitation in investigative or intelligence contexts. Coreference Resolution groups together mentions in text that refer to the same entity, traditionally within a single document but also in cross-document settings, but do not consider graph structure [12], [13].

The term Canonicalization is used specifically in the context of Open KGs. It involves clustering all mentions that refer to the same entity. The term was first introduced by Gallaraga et al. [7] from the observation that one important limitation of Open KG is that nodes are mentions that were not converted to a canonical form. This is typically approached as a clustering problem. The distinction between these tasks is often blurred, and techniques overlap. External KBs have been used for canonicalization, and coreference resolution can feed into entity linking. However, Open KG canonicalization faces unique challenges due to the form–meaning ambiguity. A single form may refer to different entities (Obama could denote Barack or Michelle) and over-aggressive canonicalization can merge distinct entities, degrading KG quality. Given these constraints, complete deduplication in Open KGs is rarely achievable. The practical objective is to design methods that balance precision and recall according to the operational context. In our case, high-stakes intelligence scenarios where false merges are more damaging than missed merges. This motivates a soft canonicalization approach: merging mentions only when all their occurrences can be confidently assigned to the same real-world entity. This strategy limits the risk of erroneous merges but introduces challenges when dealing with homonymy, that is to say cases where identical surface forms refer to multiple entities.

C. From traditional to soft Canonicalization

Various NLP and AI techniques have been applied to deduplication problems. We narrow our literature review to the canonicalization of entities in Knowledge Graphs, while still mentioning related work on deduplication in other contexts. Most canonicalization methods are formulated as a clustering problem: a pairwise similarity function is defined to compare entities, and an algorithm, most often Hierarchical Agglomerative Clustering (HAC), is used to group mentions into clusters. The main differences between approaches lie in the features used to compute the similarity metric. In the original work on canonicalization [7], several text-based features such as TF–IDF scores or Jaccard similarity were combined into a single similarity value. These methods rely solely on surface form.

Building on this intuition, later work sought to use additional information to train mention embeddings. This notion of side information is central to CESI [4]. Side information describes the context in which the relations were extracted, and includes entity linking, synonym extraction, word sense disambiguation, IDF token overlap, and morphological information to train task-specific embeddings. Other architectures, such as COMBO, relied more heavily on pretrained word embeddings. These approaches can link related mentions even when their surface forms differ completely. However, these embedding-based approaches still generally ignore the structure of the Knowledge Graph itself and fails to differentiate closely related entities (Barack Obama and Michelle Obama). Jiang et al. proposed a GNN-based approach that leverages KG connections as well as document co-occurrences [14]. It is also possible to combine multiple features such as, Shen et

al. [15] use both graph and textual features to learn mention embeddings. Liu et al. [16] went further by jointly training embeddings and clustering.

These supervised approaches are difficult to train, require numerous features that are not always available or relevant (e.g., links to external KBs), and tend to be highly domain-specific. The reported F1 scores (around 0.8 on COMBO [6] and 0.7 on ReVerb45K) are insufficient for real-world applications, especially in high-stakes contexts where precision is prioritized over recall. These limits stem in part from the clustering paradigm itself: difficulty in determining optimal stopping criteria, no mechanism to edit raw clustering results after embedding pretraining, and inherent fuzziness in vector comparisons. Errors typically occur near the boundaries of the embedding space, and optimal similarity thresholds vary from one corpus to another. Without additional constraints, nothing can be done to improve precision without severely degrading recall. To address these issues, we can draw inspiration from related domains such as record deduplication, where the problem is typically handled as a pipeline rather than a single-step clustering process. For example, the Magellan system [17] decomposes deduplication into distinct phases Blocking – a fast, inexpensive heuristic to eliminate clearly unrelated entity pairs, reducing the number of comparisons and avoiding $O(n^2)$ complexity, Entity matching – identifying pairs that refer to the same real-world entity using richer heuristics, and Entity clustering – merging validated matches into canonical entities, with an opportunity to correct inconsistencies. In contrast, most canonicalization methods for Open KGs treat the task as matching via clustering only, omitting these post-processing stages.

Since their broad release in 2022, LLMs have renewed interest in KG construction, particularly in Open IE [1], but have not been applied directly to canonicalization as we define it. Zhang et al. [18], for instance, addressed relation canonicalization by prompting an LLM to generate relation definitions and then mapping or creating triples accordingly. For entities, LLM-based pipelines usually follow an extract–refine strategy: (1) retrieve duplicate candidates via vector similarity, and (2) prompt the LLM to decide if pairs denote the same entity (e.g., EntGPT [19]). This approach could, in principle, replace the clustering phase in canonicalization pipelines, yet several barriers remain: the high cost of prompting over millions of entities, large token usage due to context requirements, and weak performance on difficult cases such as initials, short aliases, or domain-specific distinctions. More recent agentic architectures combine textual and graph reasoning for tasks like claim verification [20], and could be adapted to canonicalization by orchestrating retrieval, comparison, and decision-making. However, their computational overhead renders them impractical for large-scale knowledge graphs containing millions of entities. Viewing canonicalization as a multi-step pipeline rather than a monolithic clustering task allows heterogeneous methods to be combined. An LLM stage placed at the end can correct the instability of vector similarity while keeping usage targeted and affordable. The challenge is to design robust heuristics that filter easy cases early and reserve only borderline pairs for LLM judgment, thus combining the scalability of symbolic or embedding-based methods with the nuanced reasoning of LLMs. In contrast to SoTA approaches that either trade precision for recall or rely on costly, resource-heavy pipelines, our approach proposes a frugal, explainable hybrid architecture that integrates rule-based NLP, vector similarity, graph features, and selective LLM calls to deliver high precision without prohibitive costs.

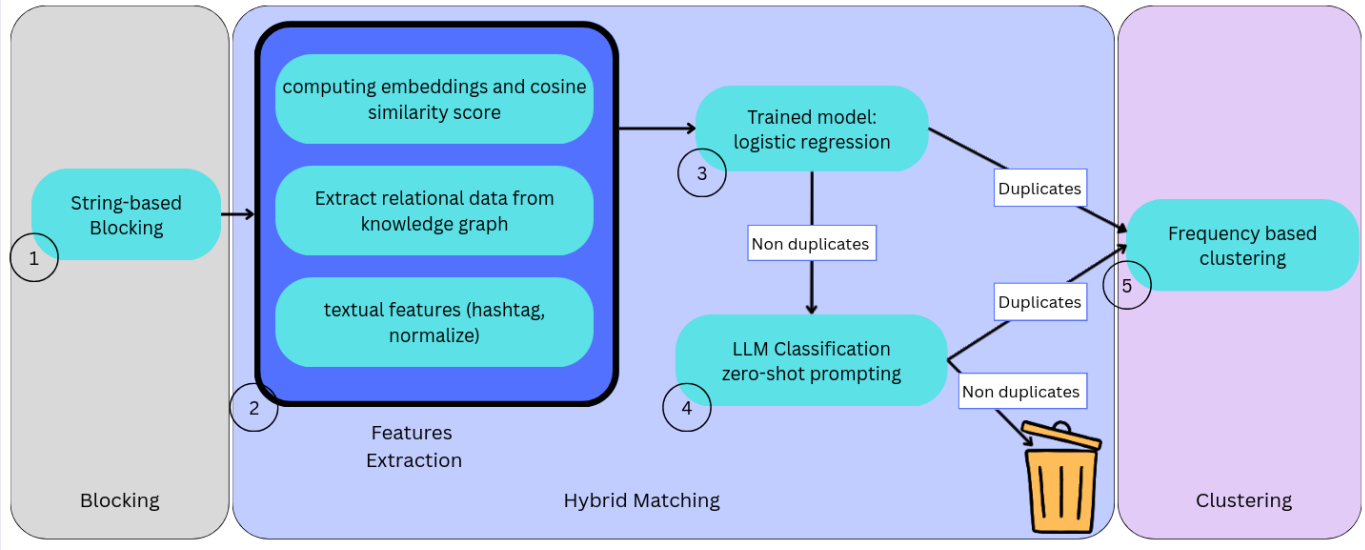


Fig. 1: **Proposed Hybrid Canonicalization Architecture.** The pipeline combines form-based blocking, explainable supervised matching, and selective LLM classification.

III. Methodology: A Hybrid Canonicalization Pipeline

Our pipeline adopts the classical three-step deduplication process (Blocking/Indexing → Matching → Clustering) while exploiting the graph structure and textual contexts of an open Knowledge Base to achieve soft canonicalization. Candidate duplicates are first retrieved through string-based similarity (blocking phase), narrowing the search space. Matching then proceeds in two stages: a lightweight supervised model classifies pairs using linguistic and relational features, and an LLM is invoked only on borderline cases to improve recall while keeping costs manageable. Finally, clustering aggregates duplicates transitively and resolves conflicts by retaining the most frequent surface form. The result is a dictionary mapping each entity to its duplicates in the knowledge graph, combining scalability with precision through the hybrid use of symbolic, statistical, and LLM-based methods.

A. Blocking Phase

Finding candidate pairs is a quadratic complexity problem. A blocking phase quickly filters out obvious non-duplicates using inexpensive string similarity metrics. At this stage, only entity names are available, so blocking relies on surface-form comparisons such as Levenshtein distance, which measures the number of edits needed to transform one string into another (e.g., Vladimir Poutine vs. Vladmir Poutine) or Jaccard similarity, which compares the overlap of n-grams between two strings making it useful for noisy or partially rearranged forms (e.g., company names).

B. Hybrid Matching phase

1) Feature extraction: For each candidate pair, three categories of features are extracted.

First linguistic features, derived from textual contexts using simple NLP rules: a boolean indicating whether one mention is a hashtag or Twitter handle variant of the other (e.g. Emmanuel Macron vs #EmmanuelMacron or @emmanuelmacron); a boolean checking if the names match once normalized by removing diacritics, dashes, special characters, and case differences (e.g. Christine Dugoin-clement vs Christine Dugoin Clément, both reduced to christine dugoin clement); and a Levenshtein similarity score giving the edit distance between the two strings.

Second, a cosine similarity feature is computed from contextual embeddings. Up to five sentences containing each entity (50–300 tokens each) are retrieved for both the original and candidate. Each context is encoded using a sentence encoder to capture its semantic meaning. A cosine similarity matrix is then computed by comparing the embeddings of the original contexts with those of the candidate contexts. The mean similarity is then calculated to obtain an overall cosine similarity score for the pair.

Finally, graph-based features are derived from the Knowledge Graph. Because nodes with overlapping neighbors are more likely to be duplicates, while isolated nodes are often errors or typos, we measure (i) the shortest path length between the two nodes, e.g. Vladimir Poutine and Vladimimr Poutine connected via Menace Nucléaire yield a path length of 1; and (ii) the number of distinct shortest paths connecting them. High-cardinality nodes with little informational value, such as Ukraine in a war corpus, can connect almost every entity, so paths passing through them are filtered out. Together, these linguistic, embedding-based, and graph-derived features provide a rich representation for distinguishing duplicates from non-duplicates.

2) supervised binary classification: These features are used to train a lightweight binary classifier, using a manually annotated subset with a train/validation/test split to tune the threshold for maximum precision. Each pair's features are fed to the model, which identifies duplicates while favoring precision over recall.

3) LLM prompted as a classifier : Pairs classified as non-duplicates can then be deferred to an LLM for secondary validation. This two-stage process reduces LLM calls and cost while preserving explainability through the use of an interpretable model. The LLM is used via zero-shot prompting [21] with a prompt containing a description of the classification task, the entities and their contexts and guidelines to guide the model on what constitutes relevant information while identifying duplicate entities. The LLM must output a decision (boolean) as well as a justification.

C. Clustering phrase

Although the term clustering is standard in the deduplication literature, it does not fully reflect our approach, which does not rely on distance thresholds or clustering algorithms. Instead, we apply a frequency-based reorganization. For each duplicate pair, the most frequent form in the database is chosen as the canonical "original." Pairs are processed in order of decreasing frequency, and duplicates are assigned according to two simple rules : Single original rule. A duplicate can only map to one original. If multiple candidates exist, the most frequent form is selected. Transitivity rule. If an entity has already been designated as a duplicate, all of its duplicates are recursively reassigned to the same original. This procedure ensures consistency and favors the most common variants as canonical forms.

TABLE I: Input / Output of the clustering phase

Input: Candidate pairs	**Output: Clusters**
Original: Volodymyr Zelensky Candidate: Volodymyr Zelenski	Original: Volodymyr Zelensky Candidates: Volodymyr Zelenski, Volodymyr Zelenskx, Zelenski, Volodymyr.Zelenski
Original: Volodymyr Zelensky Candidate: Volodymyr Zelenskx	
Original: Volodymyr Zelensky Candidate: Volodymyr.Zelenski	
Original: Volodymyr Zelensky Candidate: Zelenski	

IV. Results and Evaluation

A. The UkrainIA Corpus

Existing deduplication benchmarks proved unsuitable for our objectives: structured-record datasets lack textual contexts, canonicalization corpora are designed for clustering, and most resources assume relatively clean data, unlike the noisy real-world material we target. We therefore built a corpus of 180,000 French-language documents on the war in Ukraine, indexed in Elasticsearch. Sources include press articles, wire dispatches, web-crawled blogs, and speech-to-text transcripts of podcasts, with some OCR-derived text. All documents were enriched using a hybrid NER system (rules, trained models, and LLMs), then processed with open information extraction to populate a relation index forming our knowledge graph. All the relations are of the form (Entity A, Relation, Entity B) with entities being NP groups and Relation Verbal expressions. More than 6 million entities and concepts were extracted. Our study focuses on person entities: 134,000 nodes overall, with experiments limited to the 1,000 most frequent. Inspection showed duplicates largely stem from orthographic variants due to typos, inconsistent transliterations between Cyrillic and Latin, and OCR or speech-to-text errors.

We make available two datasets to facilitate replication and further research. First, we publish a set of 2,763 candidate pairs, each represented by the extracted features and the corresponding decision assigned by our classifier (i.e., model predictions rather than human annotations), available at: https://huggingface.co/datasets/Jourdain/Ukrainia_dataset_duplicates. Second, we release the manually annotated corpus used to train the classifier, which contains 400 pairs labeled manually.

B. Technical stack and implementation choices

For the blocking phase, we relied on Elasticsearch's native fuzzy search, which adapts the Levenshtein distance to return all strings within a maximum edit distance of 2 from a query. Unlike Jaccard-based similarity, it requires no custom analyzers and is particularly effective for spotting orthographic variants, making it well suited to generate candidate duplicates. For each candidate pair of entities, we computed a feature vector $\mathbf{x} \in \mathbb{R}^6$ that captures both lexical and structural signals of potential duplication. The following features were extracted:

1) **Contextual similarity:** A real-valued score in $[0, 1]$ representing the mean cosine similarity between the context sentences of the two entities. For each entity, up to five context sentences were considered, yielding at most 25 pairwise comparisons. or sentence embeddings, we used the BGE-M3 model [1], which is optimized for sentence similarity and demonstrates robustness across domains.
2) **Hashtag variation:** A binary indicator ($1 =$ true, $0 =$ false) denoting whether one entity is a hashtag variant of the other.
3) **Normalized form match:** A binary indicator specifying whether the normalized string representations of the entities are identical.
4) **String edit distance:** An integer in $\{1, 2\}$ corresponding to the Levenshtein distance between the two entity strings.
5) **Shortest path length:** The length of the shortest path between the two entities in the underlying knowledge graph.
6) **Path multiplicity:** The number of distinct shortest paths of the above length observed between the entities.

Each candidate pair is thus represented as a six-dimensional feature vector $(x_1, x_2, \ldots, x_6)$, which serves as input to the downstream duplicate detection model.

The first stage of the matching pipeline employed a logistic regression classifier trained on 400 manually annotated entity pairs (90% duplicates, 10% non-duplicates). Since the original corpus contained far fewer non-duplicates, we deliberately oversampled them during annotation in order to improve their detection, while still retaining a strong class imbalance that reflects the real data distribution.

Model selection and threshold tuning were performed via ten-fold cross-validation, with each fold partitioned into 60% training, 20% validation, and 20% testing subsets. For each fold, the decision threshold maximizing validation performance was recorded, yielding ten candidate thresholds. We adopted the median of these thresholds (0.9581) as the operating point for the classifier, and additionally recorded the third-quartile threshold (0.9766) for later comparison. An overview of the feature weights in the model is provided in Appendix B. For the second step of matching, we deployed an on-premise Llama 3.1 8B model [22] in a zero-shot binary classification setup (see Appendix A for the prompt).

C. Evaluation

In standard deduplication benchmarks, precision is the primary metric, but two characteristics of our dataset require nuance. First, while most benchmarks are dominated by non-duplicates, our blocking strategy yields candidate pairs that are mostly true duplicates. Reporting only precision would thus mask false merges, so we also report recall for the non-duplicate class as a better indicator of false positives. Second, since we did not annotate all duplicates for the 1,000 most frequent person entities (which would require annotating all person entities in the base), overall recall largely reflects blocking selectivity rather than the pipeline's true recall, and F-scores must be interpreted with caution.

Blocking produced 2,763 candidate pairs. Logistic regression classified 80% (2,223) as duplicates, forwarding the remaining 20% to the LLM. We evaluated (i) each matching step independently, with logistic regression using the median threshold, (ii) the full hybrid pipeline at both the median and third-quartile thresholds, and (iii) a baseline relying exclusively on LLM classification. We also conducted an ablation study to assess the contribution of each feature in the logistic regression model. The detailed results are provided

[1] https://huggingface.co/BAAI/bge-m3

TABLE II: Performance of matching steps, hybrid pipeline, and full LLM pipeline. Metrics are reported separately for the duplicate and non-duplicate classes. Support = number of annotated pairs per class. LLM calls = number of candidate pairs requiring LLM evaluation.

Setting	Class	Precision	Recall	F1-score	Support	LLM calls
Step 1: Logistic regression	Non-dup	0.189	**0.944**	0.315	107	–
	Dup	**0.997**	0.837	0.910	2656	
Step 2: LLM classification	Non-dup	0.456	**0.980**	0.623	101	534
	Dup	0.994	0.728	0.840	433	
Hybrid pipeline (median threshold)	Non-dup	0.456	0.925	0.611	107	534
	Dup	**0.997**	0.956	0.976	2656	
Hybrid pipeline (3rd quartile)	Non-dup	0.398	0.963	0.564	107	835
	Dup	**0.998**	0.942	0.969	2656	
Full LLM pipeline	Non-dup	0.237	**0.981**	0.382	107	2763
	Dup	**0.999**	0.873	0.932	2656	

in Appendix C. On our corpus, the LLM alone reached 0.999 precision on duplicates and 0.981 recall on non-duplicates, producing only two false positives and confirming the strength of such models for this task. Using a hybrid approach enabled us to successfully detect 315 duplicates that would have been missed using the classifier only. The hybrid pipeline achieved comparable precision (0.997–0.998). See Appendix D for an analysis of false positives. This balance demonstrates the value of a modular approach that preserves high precision while keeping computation under control. In terms of cost, our pipeline makes up to 5 times fewer LLM calls than the full LLM pipeline and runs 2.3 times faster, placing it ahead in terms of cost and explainability constraints (see Appendix E).

Beyond evaluation metrics, it is important to emphasize the operational significance of our results. Out of approximately 134,000 person entities in the corpus, the system detected and merged 2,538 duplicate entities into the final graph, while missing only 118 duplicates. The number of false merges remained very low (8 cases in total), and all arose in situations of high contextual ambiguity that would have challenged even professional human analysts. Notably, three of these errors were traced back to ambiguities already present in the source documents themselves (see Appendix D).

The system demonstrated robustness in fine-grained distinctions. For example, it correctly identified that the typo *Alexandre Jousset* referred to the investigative journalist *Alexandra Jousset* (author of a documentary on the Wagner Group), while also distinguishing *Alexandre Rousset*, a political journalist at *Les Échos*. Similarly, *Alexander Nemenov* was correctly paired with *Alexandre Nemenov* and successfully distinguished from *Alexander Nemov*.

The approach proved particularly valuable on the UkrainIA corpus, which contains a large number of transcriptions. In this context, the system successfully merged 39 distinct orthographic variants of *Volodymyr Zelensky*, thus normalizing orthography in 1006 different documents that mentioned him. Such deduplication has immediate benefits for downstream tasks. For instance, when applied in a Graph-RAG framework, entity normalization ensures that information is correctly consolidated, thereby preventing artificial gaps in graph connectivity and improving the reliability of retrieval and reasoning processes.

V. Limitations and Perspectives

A. Limitations

While the proposed canonicalization pipeline achieved robust results on the UkrainIA corpus, several limitations remain: While the results obtained on our corpus are very encouraging, further work is needed to assess the robustness of the method beyond the top 1,000 person entities and to confirm its applicability to other entity types such as organizations, locations, or abstract concepts. The approach could also be tested on different corpora, whether in intelligence contexts or in other domains. Initial experiments could reuse the classifier trained on the Ukraine corpus, followed by domain-adapted classifiers to evaluate whether comparable performance can be achieved across settings.

In this study, we were able to evaluate only the matching phase of the pipeline. As a result, we cannot provide a complete assessment of recall, defined as the proportion of correct pairs detected and merged by the pipeline relative to all mergeable entities in the KG (i.e., all cases where every textual occurrence of both terms refers to the same real-world entity). With more than one million entities, manually annotating the entire corpus was infeasible, and we were unable to comprehensively assess the performance of the blocking phase.

We deliberately limited blocking to form-similarity features for several reasons. While sentence embeddings of an entity's context proved highly effective in the matching phase, we chose not to use them in blocking. Entity-only embeddings such as AlBERT [23] capture surface form but fail to represent the entity's meaning across multiple contexts. We experimented with token-based embeddings (CamemBERT) [24] by max- and mean-pooling over the entity tokens, but cosine similarity between these vectors failed to produce a clear separation

between correct and incorrect pairs. In contrast, sentence-based embeddings, which represent the contexts in which an entity appears, yielded better results. However, reusing them in blocking would have been redundant with their role in matching. Sentence embeddings also introduce practical limitations:

- A sentence often contains multiple entities, which can dilute the signal for the entity of interest.
- When comparing contexts for entities x and y, all combinations ($x \times y$) must be considered, leading to a combinatorial explosion in candidate pairs.
- This shifts the problem into scenarios where incorrect pairs vastly outnumber correct ones, making matching more difficult.

By relying solely on form-based blocking, we risk missing candidates with highly dissimilar surface forms (e.g., "Obama" vs. "the 44th President of the United States"). Future work could investigate entity-based embeddings or KG node embeddings to address this gap. Currently, blocking relies on Levenshtein distance, efficiently implemented as "fuzzy search" in Elasticsearch. Other form-similarity metrics, such as Jaccard similarity or TF–IDF scores, show promise for expanding candidate lists. The challenge is to integrate these metrics efficiently into the search engine, avoiding the O(n2) complexity that arises in the worst case while ensuring fast execution.

In this work, soft canonicalization was used primarily as a normalization process: grouping textual variants of the same entity and removing redundant mentions caused by dataset imperfections. Many academic studies on canonicalization start from clean corpora without such noise. By contrast, our focus was on merges with high operational impact, prioritizing scalability and applicability in real-world, imperfect datasets. This aligns with the distinction in structured data deduplication between clean entity resolution and "dirty" entity resolution, where the latter explicitly accounts for noisy and incomplete data. Traditional canonicalization largely targets clean KBs; our approach adapts these techniques to messy, large-scale intelligence scenarios.

B. Perspective for future works

The architecture we introduced is highly modular, allowing refinement of the candidate selection phase, the addition of new features, or experimentation with alternative model architectures. It does, however, require a minimal set of annotated pairs to tune the matching model. In this respect, our design philosophy follows the approach of Magellan for structured record deduplication, offering a pipeline that data scientists can adapt and iterate on to suit both their deduplication objectives and the characteristics of their dataset.

As we have shown, training the regression model on only a few hundred annotated pairs yields robust performance on corpora similar to the training data. However, we did not have the opportunity to test its stability under significant domain shifts. In practice, parameter adjustments may be needed when the nature of the data changes substantially. We therefore advocate treating canonicalization not as a one-off operation, but as an iterative process. The removal of some duplicates can reveal new connections that point to further merges, and different deduplication settings can target distinct types of duplicates. Keeping a human in the loop, both for validation and for strategic guidance, ensures the process remains aligned with operational needs while requiring minimal annotation effort during data exploration.

This work also needs to be viewed in the broader context of Knowledge Graph (KG) construction. Canonicalization is often considered a downstream "normalization" step, but the duplication problem often originates earlier, during Open Information Extraction and cross-document coreference resolution. Our pipeline acts as a corrective rather than a preventative measure. In principle, the same techniques could be applied upstream, integrating a linking phase into KG construction so that new entities are matched against existing graph nodes as they are created. This, however, carries a risk: early in the process, when the graph is still sparse, errors in linking could propagate and be harder to detect. It is therefore possible that batch canonicalization at a later stage yields better results than fully online integration during KG building.

Finally, the boundary between KG-level and database-level deduplication is often artificial. Duplicates that appear in the graph frequently mirror duplicates in the underlying data from which it was built. Addressing both levels can improve overall data quality, with canonicalization serving as a form of textual normalization. As highlighted by Mel Richey [2], there is an ongoing debate between graph-based entity resolution and dataset-level entity resolution. The highest-fidelity graphs typically require both:

1) Resolving entities within and across datasets before they are ingested into the graph minimizes large-scale, computationally expensive post-processing.
2) Ensuring that only resolved entities enter the graph reduces operational costs while improving downstream analytics.

Our contribution shows that modular, frugal, and explainable canonicalization methods can play a role in both corrective and preventive strategies, and that operational intelligence contexts stand to benefit from embedding these methods into the KG lifecycle

VI. Conclusion

We introduced a modular pipeline for soft canonicalization in noisy knowledge graphs, combining classical similarity metrics, graph-based heuristics, and selective LLM usage. Applied to the UkrainIA corpus, our approach demonstrated that high precision can be achieved at scale while drastically reducing reliance on costly LLM calls. By separating blocking, matching, and clustering into interpretable steps, the system remains both frugal and explainable, allowing analysts to

[2]https://towardsdatascience.com/entity-resolved-knowledge-graphs-6b22c09a1442

trust and adjust its decisions. Beyond this case study, the pipeline provides a flexible foundation for canonicalization in intelligence contexts, where data is heterogeneous, imperfect, and high-stakes. Future work will extend evaluation to broader entity types and domains, and explore upstream integration of canonicalization into the knowledge graph construction process.

REFERENCES

[1] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen, "Large Language Models for Generative Information Extraction: A Survey," Oct. 2024, arXiv:2312.17617 [cs]. [Online]. Available: http://arxiv.org/abs/2312.17617

[2] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, and S. Tang, "Graph Retrieval-Augmented Generation: A Survey," Sep. 2024, arXiv:2408.08921 [cs]. [Online]. Available: http://arxiv.org/abs/2408.08921

[3] R. Fang and L. Cui, "Knowledge Graph-based Intelligence Analysis of Foreign Military Forces," in *Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering*, ser. ICAICE '23. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 518–522. [Online]. Available: https://doi.org/10.1145/3652628.3652715

[4] S. Vashishth, P. Jain, and P. Talukdar, "CESI: Canonicalizing Open Knowledge Bases using Embeddings and Side Information," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 2018, pp. 1317–1327, arXiv:1902.00172 [cs]. [Online]. Available: http://arxiv.org/abs/1902.00172

[5] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying Large Language Models and Knowledge Graphs: A Roadmap," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024, arXiv:2306.08302 [cs]. [Online]. Available: http://arxiv.org/abs/2306.08302

[6] C. Jiang, Y. Jiang, W. Wu, Y. Zheng, P. Xie, and K. Tu, "COMBO: A Complete Benchmark for Open KG Canonicalization," Feb. 2023, arXiv:2302.03905 [cs]. [Online]. Available: http://arxiv.org/abs/2302.03905

[7] L. Galárraga, G. Heitz, K. Murphy, and F. M. Suchanek, "Canonicalizing Open Knowledge Bases," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 1679–1688. [Online]. Available: https://doi.org/10.1145/2661829.2662073

[8] P. Liu, W. Gao, W. Dong, L. Ai, Z. Gong, S. Huang, Z. Li, E. Hoque, J. Hirschberg, and Y. Zhang, "A Survey on Open Information Extraction from Rule-based Model to Large Language Model," Oct. 2024, arXiv:2208.08690 [cs]. [Online]. Available: http://arxiv.org/abs/2208.08690

[9] C. Lopez, S. Verdy, G. Gadek, M. Prieur, D. Schwab, G. Sérasset, and N. Vuth, "POPCORN : IA d'extraction d'information à partir de sources textuelles pour le renseignement militaire," in *6th Conference on Artificial Intelligence for Defense*. Rennes, France: AMIAD, Nov. 2024. [Online]. Available: https://hal.science/hal-05046641

[10] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis, "End-to-End Entity Resolution for Big Data: A Survey," Aug. 2020, arXiv:1905.06397 [cs]. [Online]. Available: http://arxiv.org/abs/1905.06397

[11] E. Huaman, E. Kärle, and D. Fensel, "Duplication Detection in Knowledge Graphs: Literature and Tools," Apr. 2020, arXiv:2004.08257 [cs]. [Online]. Available: http://arxiv.org/abs/2004.08257

[12] G. Martinelli, E. Barba, and R. Navigli, "Maverick: Efficient and Accurate Coreference Resolution Defying Recent Trends," Jul. 2024, arXiv:2407.21489 [cs]. [Online]. Available: http://arxiv.org/abs/2407.21489

[13] Z. Dong, M. Wang, S. deng, L. Dai, J. Li, X. Liu, and R. Nong, "Cross-Document Contextual Coreference Resolution in Knowledge Graphs," Apr. 2025, arXiv:2504.05767 [cs]. [Online]. Available: http://arxiv.org/abs/2504.05767

[14] T. Jiang, T. Zhao, B. Qin, T. Liu, N. V. Chawla, and M. Jiang, "Canonicalizing Open Knowledge Bases with Multi-Layered Meta-Graph Neural Network," Jun. 2020, arXiv:2006.09610 [cs]. [Online]. Available: http://arxiv.org/abs/2006.09610

[15] W. Shen, Y. Yang, and Y. Liu, "Multi-View Clustering for Open Knowledge Base Canonicalization," Jun. 2022, arXiv:2206.11130 [cs]. [Online]. Available: http://arxiv.org/abs/2206.11130

[16] B. Liu, H. Peng, W. Zeng, X. Zhao, S. Liu, and L. Pan, "Open Knowledge Base Canonicalization with Multi-task Learning," Mar. 2024, arXiv:2403.14733 [cs]. [Online]. Available: http://arxiv.org/abs/2403.14733

[17] P. Konda, S. Das, P. Suganthan G. C., A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. Naughton, S. Prasad, G. Krishnan, R. Deep, and V. Raghavendra, "Magellan: toward building entity matching management systems," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1197–1208, Aug. 2016. [Online]. Available: https://dl.acm.org/doi/10.14778/2994509.2994535

[18] B. Zhang and H. Soh, "Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction," Oct. 2024, arXiv:2404.03868 [cs]. [Online]. Available: http://arxiv.org/abs/2404.03868

[19] Y. Ding, A. Poudel, Q. Zeng, T. Weninger, B. Veeramani, and S. Bhattacharya, "EntGPT: Entity Linking with Generative Large Language Models," May 2025, arXiv:2402.06738 [cs]. [Online]. Available: http://arxiv.org/abs/2402.06738

[20] H. Pham, T.-D. Nguyen, and K.-H. N. Bui, "Verify-in-the-Graph: Entity Disambiguation Enhancement for Complex Claim Verification with Interactive Graph Representation," May 2025, arXiv:2505.22993 [cs]. [Online]. Available: http://arxiv.org/abs/2505.22993

[21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," Jul. 2020, arXiv:2005.14165 [cs]. [Online]. Available: http://arxiv.org/abs/2005.14165

[22] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. v. d. Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. v. d. Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. d. Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi,

A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma, "The Llama 3 Herd of Models," Nov. 2024, arXiv:2407.21783 [cs]. [Online]. Available: http://arxiv.org/abs/2407.21783

[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," Feb. 2020, arXiv:1909.11942 [cs]. [Online]. Available: http://arxiv.org/abs/1909.11942

[24] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, V. d. l. Clergerie, D. Seddah, and B. Sagot, "CamemBERT: a Tasty French Language Model," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7203–7219, arXiv:1911.03894 [cs]. [Online]. Available: http://arxiv.org/abs/1911.03894

APPENDIX A
PROMPT TEMPLATE FOR ENTITY RESOLUTION

```
<prompt>
  <task>
    This is a task of entity resolution. You must determine whether two
    named entities refer to the exact same real-world entity.
  </task>

  <guidelines>
    You must base your decision not only on the names, but also on
    contextual information:
    <criteria>
      <item>Location</item>
      <item>Organization or affiliation</item>
      <item>Occupation or role</item>
      <item>Dates and events mentioned</item>
    </criteria>
  </guidelines>

  <warnings>
    <warning>
      Be careful: the entity names are often very similar but may contain
      typos, abbreviations, or small spelling differences.
    </warning>
    <warning>
      Be careful: two entities that have the same role but do not belong
      to the same country do not refer to the same real-world entity.
      For example, the French Air Force and the British Air Force do not
      refer to the same real-world entity, even if they have a similar
      role in their respective countries.
    </warning>
  </warnings>

  <entities>
    <entityA>
      <name>{{ENTITY_A_REPLACE}}</name>
      <contexts>{{CONTEXTS_A_REPLACE}}</contexts>
    </entityA>
    <entityB>
      <name>{{ENTITY_B_REPLACE}}</name>
      <contexts>{{CONTEXTS_B_REPLACE}}</contexts>
    </entityB>
  </entities>

  <outputInstruction>
    Respond ONLY with a JSON object matching the expected schema.
    <responseRules>
      <rule>If you are certain the two entities refer to the same
      real-world entity, set "reponse" to true.</rule>
      <rule>If you are certain they are different, set "reponse" to
      false.</rule>
      <rule>If you are uncertain or the information is insufficient,
      set "reponse" to false.</rule>
    </responseRules>
  </outputInstruction>
</prompt>
```

APPENDIX B
LOGISTIC REGRESSION WEIGHTS FOR DUPLICATE DETECTION

This annex presents the explainability analysis of the features used by the logistic regression classifier in the first matching step. Coefficients are shown with sign-aware highlighting: positive (green) values increase the probability that a pair is a duplicate, while negative (red) values decrease it. Magnitudes are not directly comparable to raw feature scales; they should be interpreted given feature standardization and the logistic link.[3]

TABLE III: Logistic regression feature weights with explanations (green = promotes merge, red = discourages merge)

Feature	Description	Weight	Explanation
Cosine similarity	Cosine similarity between contextual embeddings of the original and the candidate	**1.669200**	Dominant positive signal: higher contextual similarity strongly increases the likelihood of a true duplicate.
Nb of shortest paths	Number of distinct shortest paths in the KG between the two nodes	0.521182	Multiple short connective routes suggest shared neighborhoods; increases merge likelihood.
Hashtag	Indicator that the candidate is a hashtag/handle variant of the original	0.321581	Social-media style variants (#name, @name) frequently denote the same entity; mildly promotes merging.
Normalized	Indicator that both strings match after normalization (diacritics, case, dashes, punctuation)	0.207721	Post-normalization identity supports merging, especially for orthographic noise.
Levenshtein "distance" proxy	Number of character edits required for the strings to match	−0.428570	More edits imply farther surface forms; discourages merging unless context compensates.
Shortest path length	Number of nodes on the shortest path between the two entities in the KG	−0.571087	Longer paths indicate weak proximity; reduces merge likelihood (helps curb graph-based false positives).

Detailed analysis. Cosine similarity is the decisive differentiator: when contextual evidence is strong, the model assigns a high log-odds to a merge decision, often overcoming small spelling or tokenization differences. Graph signals refine that decision. A larger *number of shortest paths* (for fixed path quality) suggests dense shared neighborhoods, which frequently arise for co-mentioned persons in recurring stories; this boosts confidence without requiring direct string similarity. Conversely, longer *shortest path lengths* penalize merges by indicating sparse or indirect connections; this is especially useful to avoid "hub" pitfalls where high-degree nodes (e.g., *Ukraine* in a Ukraine-war corpus) could otherwise spuriously connect unrelated mentions.

Linguistic features act as pragmatic tie-breakers. The *Hashtag* indicator captures a common aliasing pattern in OSINT and media data (#EmmanuelMacron, @emmanuelmacron), which provides weak but reliable evidence. The *Normalized* match feature counters OCR/ASR and typographic noise by collapsing diacritics, punctuation, and case; it improves recall on near-miss variants without inflating false positives. The (inverse) *Levenshtein* signal discourages merges as edit distance grows, but its negative impact is moderated whenever contextual similarity is high, allowing the model to merge semantically close mentions that differ in transliteration or spacing.

Taken together, these coefficients implement a robust decision logic: (i) default to context (cosine) as the primary signal; (ii) use the graph to reward dense, short-range proximity and penalize tenuous links; and (iii) let inexpensive string cues arbitrate borderline cases. This division of labor explains the model's behavior on ambiguous pairs: when cosine is neutral (around 0.5), small boosts from *Normalized* or *Hashtag* can tip the scales, whereas long path lengths or large edit distances can veto merges that would otherwise be suggested by noisy context overlaps. In operational settings, this yields an interpretable, precision-oriented front end that filters most pairs before the LLM step, thereby reducing cost while preserving analyst trust.

[3]If features were standardized (zero mean, unit variance), coefficients are on a comparable scale; otherwise, interpret relative to typical feature ranges.

APPENDIX C
ABLATION STUDY ON THE CLASSIFIERS'S FEATURES

This annex reports the results of our ablation study, designed to evaluate the contribution of each feature family:

C Cosine similarity (context embeddings),
G Graph features (number and length of shortest paths),
L Linguistic features (Levenshtein similarity, hashtag, normalization).

The evaluation focuses on three metrics most relevant to our use case:

1) Precision on the **duplicate** class (high precision is critical in intelligence),
2) Recall on the **non-duplicate** class (to limit false merges),
3) Recall on the **duplicate** class (to maintain coverage).

TABLE IV: Ablation results across feature subsets. Best values are highlighted in green, worst in red.

Feature set	Dup. Precision	Non-dup. Recall	Dup. Recall
L	1.00	1.00	0.03
CGL (*our model*)	0.98	0.92	0.59
C	0.98	0.91	0.52
CG	0.98	0.91	0.47
CL	0.97	0.84	0.57
G	0.95	0.99	0.06
GL	0.95	0.96	0.12

Analysis. The full model (CGL) offers the best balance across metrics, achieving both high precision (0.98) and strong recalls (non-duplicate: 0.92, duplicate: 0.59). By contrast, feature subsets that optimize one metric do so at the expense of others:

- Using **linguistic features alone (L)** yields perfect precision but collapses duplicate recall (0.03), making it practically unusable.
- **Graph features (G)** maximize non-duplicate recall (0.99) but severely underperform on duplicate recall (0.06), reflecting their tendency to over-separate entities. The graph structure is not informative enough to enable decision.
- **Cosine similarity (C)** is the strongest standalone predictor, balancing precision and recall reasonably well, though still below the full model.
- Combining features (e.g., CL, CG) improves robustness but never outperforms the full CGL pipeline.

In short, ablation confirms that no single feature family suffices: **cosine similarity anchors the model, graph features provide structure-aware regularization, and linguistic cues resolve borderline cases.** Their combination explains why the CGL pipeline consistently outperforms weaker alternatives in high-stakes deduplication.

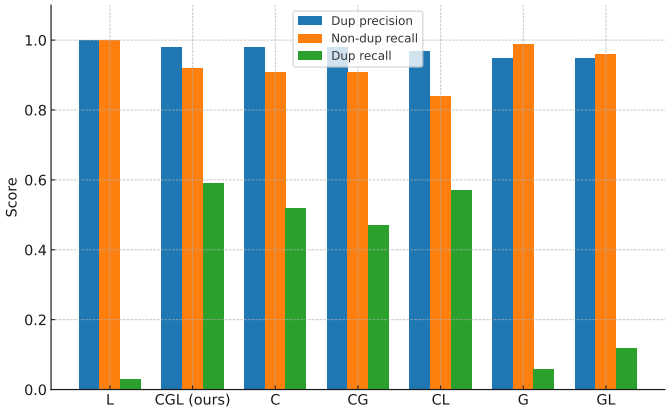


Fig. 2: Ablation study of Feature Subsets

APPENDIX D
FALSE POSITIVES: FEATURES AND ANALYSIS

This annex details the eight false positives. Table V lists the model features for each pair; Table VI provides a short human analysis explaining the confusion.

TABLE V: Feature outputs for false-positive pairs (green = favorable to merge, red = risky)

Original	Candidate	Lev.	Norm.	Hash.	Cosine	Len sp	# sp
Olivier Véran	Olivier Véron	1	False	False	0.39124	0	0
Emmanuel Macron	Emmanuel Mandon	2	False	False	0.39773	1	1
Aleksandar Vučić	Aleksandar Vulin	2	False	False	0.53017	1	4
Anne Rovan	Anne Jouan	2	False	False	0.54916	0	0
Guillaume Daret	Guillaume Garot	2	False	False	0.57206	2	4
Oleg Orlov	Oleg Perlov	2	False	False	0.58875	0	0
John Kirby	John Kerry	2	False	False	0.60165	1	2
Dmitry Peskov	Dmitry Lyskov	2	False	False	0.66919	3	0

Notes. "Len sp" = shortest-path length; "# sp" = number of distinct shortest paths. Colors are heuristic: higher cosine and path count (green) tend to promote merges; higher edit distance and long paths (red) discourage merges.

TABLE VI: "Who is who" analysis for each false-positive pair

Original	Candidate	Analysis (who is who and why confused)
Olivier Véran	Olivier Véron	**Véran**: former French Minister of Health; Government spokesperson. **Véron**: economist; the name here stems from a journalistic mistake : actual person is *Nicolas Véron*. Orthographic proximity and French political context drove the merge.
Emmanuel Macron	Emmanuel Mandon	**Macron**: President of France. **Mandon**: French MP (LREM, Macron's party). Party/role proximity inflated similarity.
Aleksandar Vučić	Aleksandar Vulin	**Vučić**: President of Serbia. **Vulin**: Deputy PM; former head of the Security Intelligence Agency. Both times under Vučić, which resulted in co-mentions and inflated similarity.
Anne Rovan	Anne Jouan	**Rovan**: French journalist, Brussels correspondent. **Jouan**: investigative journalist, author on the Mediator affair. Mentioned briefly during an emission on the Ukraine war in a sentence announcing a coming up subject on the mediator drug.
Guillaume Daret	Guillaume Garot	**Daret**: French journalist. **Garot**: French MP (Mayenne). Same first name + frequent co-mentions in French political news.
Oleg Orlov	Oleg Perlov	**Orlov**: Russian human rights activist. **Perlov**: likely *Andrey Perlov* (retired race walker). Both referenced around Russian detentions; same first name, sparse contexts.
John Kirby	John Kerry	**Kirby**: White House National Security Communications Advisor. **Kerry**: U.S. Special Presidential Envoy for Climate. Author accidentally used Kerry instead of Kirby; U.S. gov't context and similar surnames.
Dmitry Peskov	Dmitry Lyskov	**Peskov**: Kremlin Press Secretary. **Lyskov**: Head of press service for the Kaliningrad enclave. Russian government media contexts and same first name increased confusion.

These errors cluster around :

1) **role/affiliation proximity** (same party, same government, same beat)
2) **orthographic similarity** (close edit distance, same first name)
3) **noisy context windows** (brief mentions, newsroom slips)

This annex highlights two complementary error regimes:

1) **LLM false positives**, where the large language model overrode its own uncertainty or hallucinated attributes;
2) **Logistic regression false positives**, where the statistical model was misled by inflated similarity signals.

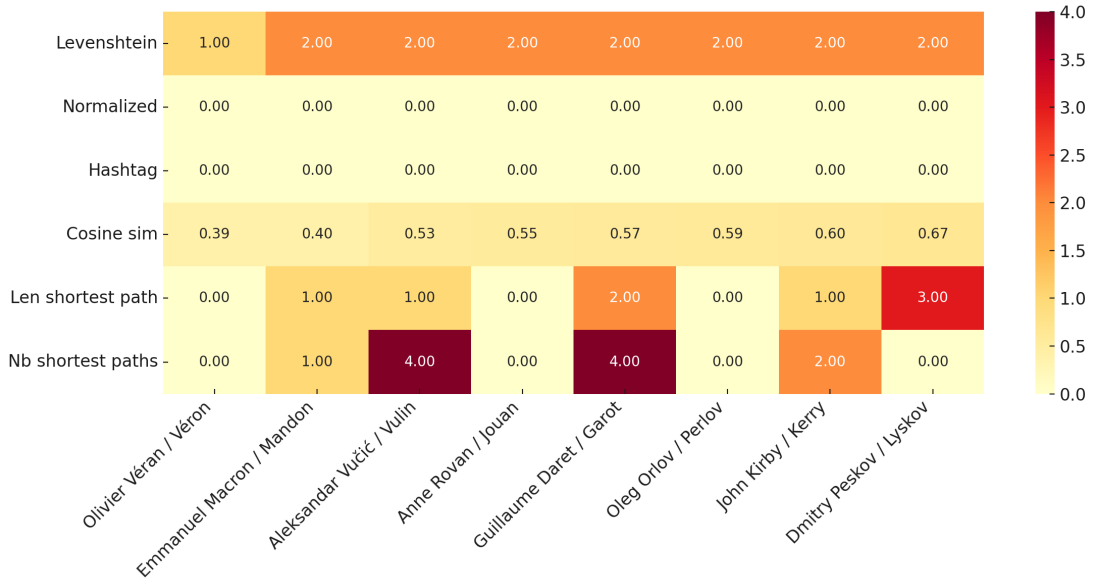


Fig. 3: Heatmap analazing the features involved in incorrect decisions

LLM False Positives

These two pairs were correctly rejected as non-duplicates by the logistic regression model during the first stage, but erroneously accepted as duplicates by the LLM.

1) Olivier Véran / Olivier Véron :
   LLM rationale: ”Both entities have the same name and occupation (porte-parole du Gouvernement), and they are mentioned in the same contexts, which suggests that they are the same person. The difference in spelling (Véran vs Véron) is likely a typo.”
   Comment: The LLM hallucinates an occupation that does not appear in the candidate’s only context. No evidence links “Olivier Véron” to a government spokesperson role.

2) Emmanuel Macron / Emmanuel Mandon :
   LLM rationale: ”Both entities have the same name, Emmanuel, and the contexts provided suggest that they are both politicians. The contexts for entity A mention Emmanuel Macron as the President of France, while the context for entity B mentions Emmanuel Mandon as a deputy. However, the names are very similar and it is possible that Emmanuel Mandon is a misspelling or a variation of Emmanuel Macron’s name. Further investigation is needed to confirm the identity of Emmanuel Mandon, but based on the information provided, it is likely that the two entities refer to the same real-world entity.”
   Comment: The LLM contradicts itself. It explicitly states that ”further investigation is needed” (thus acknowledging insufficient evidence), yet still outputs a positive label. In doing so, it violates the caution directive: *“If you are uncertain or the information is insufficient, set response to false.”*

Logistic Regression False Positives

In contrast, the logistic regression model is dominated by the cosine similarity of contextual embeddings. Graph-derived and linguistic features are designed to act as counterweights. Most false positives occur when cosine similarity is unusually high while other features fail to oppose it strongly enough.

Errors are driven by :

1) **Authorial mistakes** : contexts where journalists misattribute the same role or position to two distinct individuals.
2) **Impoverished contexts** : candidates appearing only once in a low-information sentence (e.g., name lists).
3) **Contextual overlap** : cases where each entity is cited within the other’s context window, artificially inflating relatedness.
4) **Very similar contexts** : situations where both entities share multiple attributes in their contexts, leading to elevated cosine scores.

APPENDIX E
FRUGALITY AND TOKEN COST

This annex compares the financial and runtime costs of our proposed hybrid pipeline against a full LLM pipeline that relies exclusively on LLM classification in the matching phase. All evaluations were conducted on the *1000 most frequent entities* from our soft canonicalization run. LLM costs are derived from token pricing published on `prompthub.us`, using average input/output token counts and call frequencies observed in our experiments.

5.1 Token Costs by Model

TABLE VII: Cost per million tokens (in USD) from `prompthub.us`.

Model	**Input tokens**	**Output tokens**
Llama 3.1 8b (Azure)	0.31	0.61
GPT-5	1.25	10.00

5.2 Comparative Pipeline Costs

TABLE VIII: Comparative costs of the hybrid pipeline vs a full LLM pipeline. Costs expressed in USD.

Pipeline	**Pairs**	**LLM Calls**	**Input Tokens**	**Output Tokens**	**GPT-5 Cost**	**Llama 3.1 8b Cost**
Hybrid pipeline	2763	534	694,827	51,820	1.38673	0.24006
Full LLM	2763	2763	3,597,426	270,774	7.20452	1.24440

5.3 Runtime Comparison

TABLE IX: Pipeline runtimes vs projected full LLM runtime (average of 1.135 calls/sec).

Pipeline	**Calls**	**Avg Calls/sec**	**Runtime (sec)**	**Runtime (min)**
Hybrid pipeline	534	1.135	1823	30.38
Full LLM	2763	1.135	4283	71.37

5.4 Analysis

The hybrid pipeline delivers a 5.2× reduction in token costs (from \$7.20 to \$1.38 on GPT-5), and a 2.3x runtime speedup (71.4 minutes $\rightarrow$ 30.4 minutes). This frugality comes from leveraging lightweight similarity and graph heuristics to filter out easy negatives, reserving LLM calls only for ambiguous cases at the decision frontier. In intelligence contexts, this balance of efficiency, precision, and interpretability is critical: it allows high-quality canonicalization while keeping costs bounded and throughput realistic for operational deployments.

Khiops: An End-to-End, Frugal AutoML and XAI Machine Learning Solution for Large, Multi-Table Databases

Marc Boullé, Nicolas Voisine, Bruno Guerraz, Carine Hue, Felipe Olmos, Vladimir Popescu, Stéphane Gouache, Stéphane Bouget, Alexis Bondu, Luc Aurelien Gauthier, Yassine Nair Benrekia, Fabrice Clérot, Vincent Lemaire
(Orange Research, contact: firstname.name@orange.com)

***Abstract*—Khiops is an open source machine learning tool designed for mining large multi-table databases. Khiops is based on a unique Bayesian approach that has attracted academic interest with more than 20 publications on topics such as variable selection, classification, decision trees and co-clustering. It provides a predictive measure of variable importance using discretisation models for numerical data and value clustering for categorical data. The proposed classification/regression model is a naive Bayesian classifier incorporating variable selection and weight learning. In the case of multi-table databases, it provides propositionalisation by automatically constructing aggregates. Khiops is adapted to the analysis of large databases with millions of individuals, tens of thousands of variables and hundreds of millions of records in secondary tables. It is available on many environments, both from a Python library and via a user interface.**

***Index Terms*—Khiops, AutoML, frugal, multi-table, XAI**

I. What makes Khiops different

Khiops is an end-to-end Machine Learning (AutoML) solution that natively and effortlessly handles complex and time-consuming data science tasks on multi-million instance datasets. Khiops tasks include variable engineering (A), data cleaning and encoding (B), and parsimonious model learning (C) (see Figure 1). Khiops also includes features that allow it to be fully explainable (XAI).

The AutoML capability allows Khiops to process tabular or relational data with complex star or snowflake schemas. This is a real differentiator in a variety of situations, particularly when dealing with use cases with multiple records per statistical individual (such as calls, transactions or production logs). In a world of increasingly sophisticated cyber attacks, log analysis has become a necessity. Imagine being able to identify an intrusion in real time or precisely retrace an attacker's route to limit the damage. That's exactly what effective log management can do [1], [2].

The uniqueness of Khiops lies in its different approach to typical AutoML solutions, which often run an expensive range of complex algorithms on parameter sets using grid search. Instead, Khiops uses an original formalism called MODL (which is hyperparameter-free), allowing it to push the boundaries of automation on very large multi-table datasets and push the boundaries of automation. This allows it to build high-performance models that are simple to deploy and easy to interpret. Khiops comes with a low-code Python library that offers an efficient AutoML pipeline in a simple $.fit()$ function. Its sophisticated algorithms are easy to use, thanks to its Python library that follows Scikit-learn (sklearn) standards. Khiops facilitates automatic learning in a complete safety environment. This approach significantly reduces the time spent on the modelling phase, allowing users to allocate more time to analyse their models and gain a deeper understanding of their data, while requiring minimal coding.

Khiops is equipped with an interactive visualisation tool that provides full access to the preparation and modelling results directly from a notebook or dedicated application. Consequently, there is no requirement to write specific visualisation code to present and interpret modelling results. In addition, Khiops offers a version with a graphical interface that allows all learning algorithms to be used without the need to write a single line of code, making it easily usable by business domain specialists without requiring in-depth knowledge of data analysis.

II. An original Bayesian formalism

Whether for variable creation, transformation and selection, co-clustering or decision trees, Khiops uses an original Bayesian formalism, MODL [3]. The MODL approach aims to select the most likely model given the training data. Bayes' formula is therefore the starting point for deriving the optimisation criteria used, the general form of which is as follows:

$$arg\max_{h \in \mathcal{H}} P(h|d) = arg\max_{h \in \mathcal{H}} \frac{P(h)P(d|h)}{P(d)}$$

All MODL optimisation criteria are designed in the same way (optimal coding, automatic variable engineering and parsimonious learning), according to the following steps:

- define the $\mathcal{H}$ family of models, i.e. the modelling parameters, as a function of the learning task to be performed (i.e. $\mathcal{H}$ can be a discretization [4], a grouping of values [5] or a decision tree [6]);
- define the prior distribution on these parameters $P(h)$, which is always hierarchical and uniform at each stage of the hierarchy;

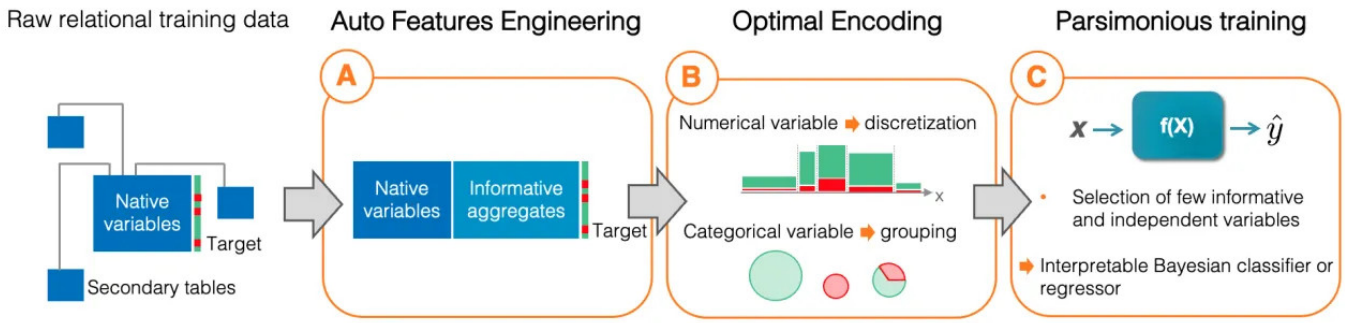


Fig. 1. Machine learning process implemented by Khiops

- obtain an optimisation criterion from the development of Bayes' formula, taking into account the likelihood term $P(d|h)$;
- learn the model by optimising the final criterion.

In information theory, the model selection problem described above can be translated into an encoding problem, the aim of which is to find the most compact way of encoding an information source for transmission over a telecommunications channel. Consider an information source emitting symbols [for example, a, b, c, etc.] whose alphabet is known. In information theory, the negative logarithm of the probability of a symbol being transmitted ($-log(P(a))$) represents its optimal coding length, denoted by L and expressed in bits. According to Shannon's intuition, the most efficient encoding strategy is to assign a short coding length to the most frequent symbols. Similarly, the probabilities in Bayes' formula above can be replaced by negative logarithms to obtain a MODL criterion to minimise, which can be interpreted as follows:

$$-log(P(h).P(d|h)) = \underbrace{L(h)}_{Prior} + \underbrace{L(d|h)}_{Likelihood}$$

- the prior corresponds to the coding length of the model, i.e. the number of bits needed to describe it;
- the likelihood is the coding length of the training data knowing the model.

In this particular instance of the encoding problem, the model is first transmitted over the telecommunications channel, followed by the data. The Minimum Description Length (MDL) principle aims to select the most compact model describing the data, and is applied in the MODL approach by choosing a hierarchical prior representing successive choices of model parameters.

III. Tool presentation

The Khiops tool integrates the work carried out at Orange Research on data preparation, automatic construction of variables for multi-table databases and large scale modelling. Since 2024, the Khiops V10 version has been open source. The very recent last version (V11) includes the following main features:

- management of multi-table data,
- automatic feature construction to generate a flat table of individuals $\times$ variables,
- automatic feature construction from text variables,
- optimal data preparation via discretisation and value grouping,
- random forests for classification and regression,
- modelling using a naive Bayesian classifier, with optimal univariate pre-processing, variable selection and learning of weights for each variable,
- deployment of models directly on multi-table bases,
- interpretation and reinforcement models,
- optimal histograms for univariate data exploration,
- variable $\times$ variable coclustering, for joint density estimation,
- instance $\times$ variable coclustering, for exploratory analysis,
- end-to-end management of sparse data,
- handling of local as well as cloud storage.

The tool is written in C++ for the algorithmic component and Java for the graphical interface. It can be used with either a graphical user interface or a Python library, allowing for easy integration into a processing pipeline.. There is also an interactive visualisation tool available for inspecting the results of preparation, modelling and evaluation (see Figure 4).

Khiops is available at http://www.khiops.org. The current version (V11) is used in a wide range of applications: including customer marketing (attrition models, appetite for new services, etc.), text mining, web mining, banking, social networks, technical and economic studies, internet traffic characterisation, ergonomics, user sociology, fraud detection, ... It has been used with learning databases containing millions of individuals and hundreds of millions of secondary records.

A. Installation :

The Khiops python library is easy to install using the conda package manager.

```
# Windows/Linux/macOS
conda install khiops -c conda-forge -c khiops
```

B. Automatic variable construction:

In the case of multi-tables, this is one of the major contributions of the tool. It is based on the description of a multi-table star or snowflake schema[1], with a root table containing the individuals to be analysed (e.g. customers) and secondary tables in 0-1 or 0-n relationships containing records completing the description of the individuals (e.g. communication details).

The only user parameter is then the number of variables to be constructed, by systematically applying selection or aggregation functions. This method used [7] exploits a Bayesian regularisation approach based on a parsimonious prior distribution over the potentially infinite set of all the variables that can be constructed. Variables are then constructed using an efficient sampling algorithm according to this prior distribution. The resulting method is simple to use, computationally efficient and robust to the problem of overlearning. The creation of MODL decision trees is the final step in the AutoML pipeline implemented by Khiops. This optional pre-processing step involves building decision trees from native variables and aggregates [6], resulting in the model as a parsimonious AutoML "random forest" model.

The important point to understand here is that the users only need to provide the schema of their data and the number of variables to be constructed, with selection and aggregation functions applied systematically. The process is fully automated. This saves a great deal of time, as there is no need to build aggregates by hand, which would require considerable business expertise. It also means that aggregates can be discovered on subjects where the Khiops user is not an expert, as well as new aggregates (new knowledge) on familiar subjects. The 'accident' use case below is clearly not an example of cyber defence, but all the principles remain valid.

C. Optimal preparation :

Data is prepared using supervised discretisation [4] for numerical variables and supervised grouping of values [5] for categorical variables. The associated methods exploit a Bayesian model selection approach to construct the most likely preparation model given the data, which provides an accurate and robust estimate of the univariate conditional density per descriptive variable.

[1]The terminology used is similar to that of data warehouses, such as star or snowflake schemas. However, here we are not talking about concepts for structuring a data warehouse, but rather about describing individuals in a statistical analysis, with some variables coming from the root table and others from secondary tables.

D. Parsimonious learning:

Modelling takes advantage of all initial variables, as well as those constructed after preparation, combining them using a naïve Bayesian classifier with variable selection and direct learning of weights per variable [8].

E. Automatic adaptation to material resources:

Khiops adapts its algorithms to the available hardware resources (RAM and CPU). Khiops divides the data into a more or less fine-grained matrix of files by partitioning the instances into rows and the variables into columns, depending on the learning task in hand and the hardware resources available. The successive stages of the AutoML pipeline are algorithms that process either rows or columns of the root table. For example, optimal encoding is a column-based algorithm, since each discretisation or clustering model can be learned independently for each variable. On the other hand, once the pipeline is executed, making predictions is a row-based algorithm, since each example can be processed independently. The aim is to optimise the execution time of these algorithms, whatever the size of the data processed and the amount of hardware resources available. Take, for example, the Zeta classification problem (9.3 GB) of the Large Scale Learning Challenge [9], which contains $500,000$ training examples and $2,000$ numerical explanatory variables. Learning on an Intel Xeon Gold 6150 2.70 Ghz processor takes 81 minutes with a single core and 512 MB of RAM, and only 3 minutes with 32 cores and 16 GB of RAM (See Section V for more details on the Zeta problem).

F. Interfaces :

Although Khiops provides a core Python library `khiops.core` to effectively meet the challenge of large volumes, it is also possible to start with the `khiops.sklearn` library for those familiar with the popular sklearn library, or even to use a GUI with Khiops Desktop. Online deployment of Khiops models for real-time applications can be done using the KNI library. Finally, it should be noted that models learned by Khiops can be easily interpreted using dedicated visualization tools.

G. Khiops is an environmentally-friendly tool (frugal [10]):

The Khiops code is highly optimised: (i) advanced optimisation algorithms have been designed specifically for each type of learning task, (ii) they have been implemented in "low-level" C++ using very fine-grained optimisation close to the hardware layer. Khiops intelligently adapts the execution of algorithms to the available hardware resources, taking into account the size of the task to be executed. The solution is compact enough to be embedded. In this way, Khiops is able to run transparently on a Rasberry, a phone, ... , with

data that far exceeds the available RAM, or on a Kubernetes cluster by adapting the number of nodes used to the size of the data. There is never any need to invest in large hardware, as execution time is the adjustment variable: 'Khiops does the best in all cases'. The models generated by Khiops adapt to the data and the machine learning task. For a simple problem, Khiops produces a parsimonious, intelligible model with few parameters, and therefore inexpensive to deploy and interpret! Khiops uses data reduction natively (parsimony): the model explicitly selects a subset of the variables and only these variables are required for deployment.

H. Khiops is an XAI tool :

As described below in the example on the 'Accidents' database Khiops also offers an interactive results visualization tool, called Khiops Visualization (figure 4). This tool allows to visualize all analysis results in an intuitive way, offering a quick and easy interpretation. This visualisation tool allow to interpret the model's global behaviour for the whole dataset. But the tool also offer the possibility to obtain local behaviour, local explanations per example. Firstly by computing the Shapley values of all the input variables of a trained classifier for each example of a deployment (or test) dataset, see [11] for more details. Secondly by suggesting variable change (univariate change) to improve (to reinforce) the probability to belong to a class on interest (in the sense of a counterfactual but where the value[2] of a single variable has been changed) see [13] (Section 4) for more details.

IV. Examples of use

A. The Accident Database

In this example, we will show how Khiops can be used to train a classifier on complex relational data where a secondary table is itself a parent table of another table (i.e. a flake schema). We will train a multi-table classifier on the Accidents dataset. The Accidents database lists all the accidents involving injuries that occurred during 2018 in France, with a simplified description.

This database includes the following information:

- The location of the accident (Places table);
- The characteristics of the accident (Accidents table);
- The vehicles involved (Vehicles table);
- The passengers in the vehicles (Users table);

The data is organised according to the following relational snowflake schema.

```
Accidents
|
| -- 1:n -- Vehicles
|              |
|              |-- 1:n -- Users
|
| -- 1:1 -- Places
```

[2]The computation of a 'complete counterfactual' will be available in 2025 on www.khiops.org as a notebook python [12].

To train the `KhiopsClassifier` with this data, we then need to specify a multi-table dataset: the main table **Accidents**, the secondary tables **Vehicles** and **Places**, the tertiary table **Users**.

1) Multi-table specification:: The first step is to specify the schema of the multi-table dataset. Khiops offers an extension to sklearn's single-table description. The main Accidents table and the secondary Places table have a single key: 'AccidentId'. The Vehicles (the secondary table) and Users (the tertiary table) tables have a key with two fields: 'AccidentId' and 'VehicleId'. To describe the relationships between the tables, the relationships field must be added to the table specification dictionary. For a $0:1$ relationship instead of $0:n$, 'True' must be added at the end of the relationship specification (see Figure 2):

```
X_accidents_train = {
    "main_table": (accidents_df.drop("Gravity", axis=1),
                   ["AccidentId"]),
    " additional_data_tables ": {
        "Vehicles": (vehicles_df, ["AccidentId", "VehicleId"]),
        "Vehicles/Users": (users_df, ["AccidentId", "VehicleId"]),
        "Places": (places_df, ["AccidentId"], True),
    },
}
y_accidents_train = accidents_df["Gravity"]
```

Fig. 2. Specification of the multi-table dataset

2) Learning: : Like a `sklearn` classification, it is simply a matter of using the functions `khc.fit` for learning and `khc.predict` for deployment (see Figure 3). In the table II, we varied $n\_features$ and max_cores to observe their influence on performance in time and AUC. We quickly noticed that increasing the number of aggregates improved performance, and that increasing the number of cores used greatly reduced analysis time.

```
# Creating a Khiops model with AUTO Feature Multi-table
khc = KhiopsClassifier(n_trees=0, n_features=10,max_cores=1)
# Train the model
khc.fit(X_accidents_train, y_accidents_train)
# Predict labels
y_pred = khc.predict(X_accidents_train)
# Calculate probabilities
y_probas = khc.predict_proba(X_accidents_train)
```

Fig. 3. Learning and deploying on the Accidents database

3) Viewing results:: Although the core api `khiops.core` contains all the methods to analyze Khiops results, Khiops also offers an interactive results visualization tool, called Khiops Visualization (figure 4). This tool allows to visualize all analysis results in an intuitive way, offering a quick and easy interpretation.

Khiops Visualization is composed of several panels. Depending on the analysis type, the panels and their contents

ProbGravityLethal	ShapleyVariable_Lethal_1	ShapleyPart_Lethal_1	ShapleyValue_Lethal_1	ShapleyVariable_Lethal_2	ShapleyPart_Lethal_2	ShapleyValue_Lethal_2
0,708149757	Max(Vehicles,Min(Users,BirthYear))	]-inf,1933,5]	0,468556017	Min(Vehicles,Min(Users,BirthYear))	]-inf,1933,5]	0,407707682
0,688394752	Max(Vehicles,Min(Users,BirthYear))	]-inf,1933,5]	0,468556017	Light	NightNoStreetLight	0,419425784
0,575788413	Light	NightNoStreetLight	0,419425784	InAgglomeration	No	0,321445857
0,548183385	Mean(Vehicles,Min(Users,BirthYear))	]-inf,1938,25]	0,363729716	InAgglomeration	No	0,321445857
0,547824738	Light	NightNoStreetLight	0,419425784	Mean(Vehicles,Min(Users,BirthYear))	]-inf,1938,25]	0,363729716

TABLE I
ILLUSTRATION OF ONE XAI OUTPUT THAT CAN BE PROVIDED BY KHIOPS.

Features number	10	100	1 000	10 000	100 000
Train AUC	0.792	0.826	0.845	0.865	0.874
Test AUC	0.781	0.818	0.838	0.855	0.854
Time with 1 core	3	8	33	273	2552
Time with 5 cores	3	4	12	76	712
Time with 9 cores	3	4	8	52	438

TABLE II
KHIOPS LEARNING PERFORMANCE ON THE ACCIDENTS TABLE ACCORDING TO THE NUMBER OF AGGREGATES GENERATED. PERFORMANCES INCLUDE AUC IN TRAIN AND IN TEST, AS WELL AS LEARNING TIME IN SECONDS FOR 1, 5, AND 9 CORES.

are not the same. In case of a supervised analysis (as for the Accident database), Khiops Visualization can be composed with 5 panels (see the top of the figure 4): (i) Preparation: displays the Preparation report; (ii) Tree preparation : displays the preparation report for tree variables; (iii) Preparation 2D: displays the 2D preparation report (iv) Modelling: displays the modelling report; (v) Evaluation: displays on one panel the test, train and evaluation reports. Finally Project Infos : displays the report file and database locations plus some short comments on the analysis. All the panels are described in a lot of details on https://khiops.org/ui-docs/visualization/

4) Variable Importance results:: To illustrate one of the XAI aspects (see section III-H) of Khiops, we give in Table I one example of the outputs it can provides. This table gives the 5 accidents among the ones with high probably of being lethal (predicted by the classifier) in the first column. We ask the tool to give for each accident the two variables which contribute the more to the predicted probability (the number of variables is just define per user when asking this XAI outcome) to be lethal. Therefore here, after the first column, there are 2 triplets of columns. Each triplet gives for each accident the name of the variable, then the value of the variable and finally the Shapley value for this variable. The triplets (so the columns of the file) are sorting according to the Shapley value allowing a fast understanding of the individual variable importances.

When examining the second accident (line 2) in this table, we see that the most important variable is "Max(Vehicles,Min(Users,BirthYear))" and the second one is "Light". The value of the most important belongs to the value interval]-inf,1993.5] while the value of the second most important value belongs to the to the categorical value "NightNoStreetLight". The associated shapley values are in columns 4 and 7. For this accident, the main causes of a high probability of being lethal are therefore easy to understand one of the vehicles involved in the accident involves an old=occupant, born before 1993 and the absence of light in the street during the night. The others lines of this table appear to be equally straightforward to read.

Of course Khiops can also output a file with all the Shapley values for all variables and for all the classes, allowing the use of this file with a python library like Shap [14] to create personalized visualisation.

B. The UNSW-NB15 dataset

In this section we follow exactly the same process than in the previous section except that we use the Khiops library on the UNSW-NB15 dataset[3] which is a flat dataset[4].

This UNSW-NB 15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours. This dataset includes nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The authors [15] have generated 49 features from the initial logs plus the class label. A partition from this dataset is configured as a training set and testing set. In this section we used this given partition. Note: When downloading the dataset from Kaggle we had only 43 features.

1) Learning: The process is very close to the one described in Section IV-A2. However, preliminary results (not presented here) show a significant covariate shift between the training and test sets. Indeed using the same methodology as in [16], we discover that the 'id' variable is the main cause of this drift and the consequences could results in a shift between train and test results. The interested reader may find on the GitHub page https://github.com/vincentlemaire-labs/CAID2025 the code to detect the drift between train and test dataset and the one to train the classifiers. A good idea could be to conduct an analysis to remove all the variables that carry the drift as in [17], but here for simplicity, and comparison purpose to past papers published on this dataset, only the 'id' variable has been removed.

2) Results: We present in Table III the results obtained with Khiops (without decision trees) as well as those obtained using other classifiers, namely Catboost (CB) [18] and Random Forest (RF) [19], both with their default parameters in scikit-learn. Note: Khiops is able to handle the UNSW-NB15 dataset

[3] https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15/data

[4] In the description of this dataset, it appears to be based on an initial star schema, but it no longer seems to be available. We have contacted the creators of the dataset to request the relational version, but we have not received a response.

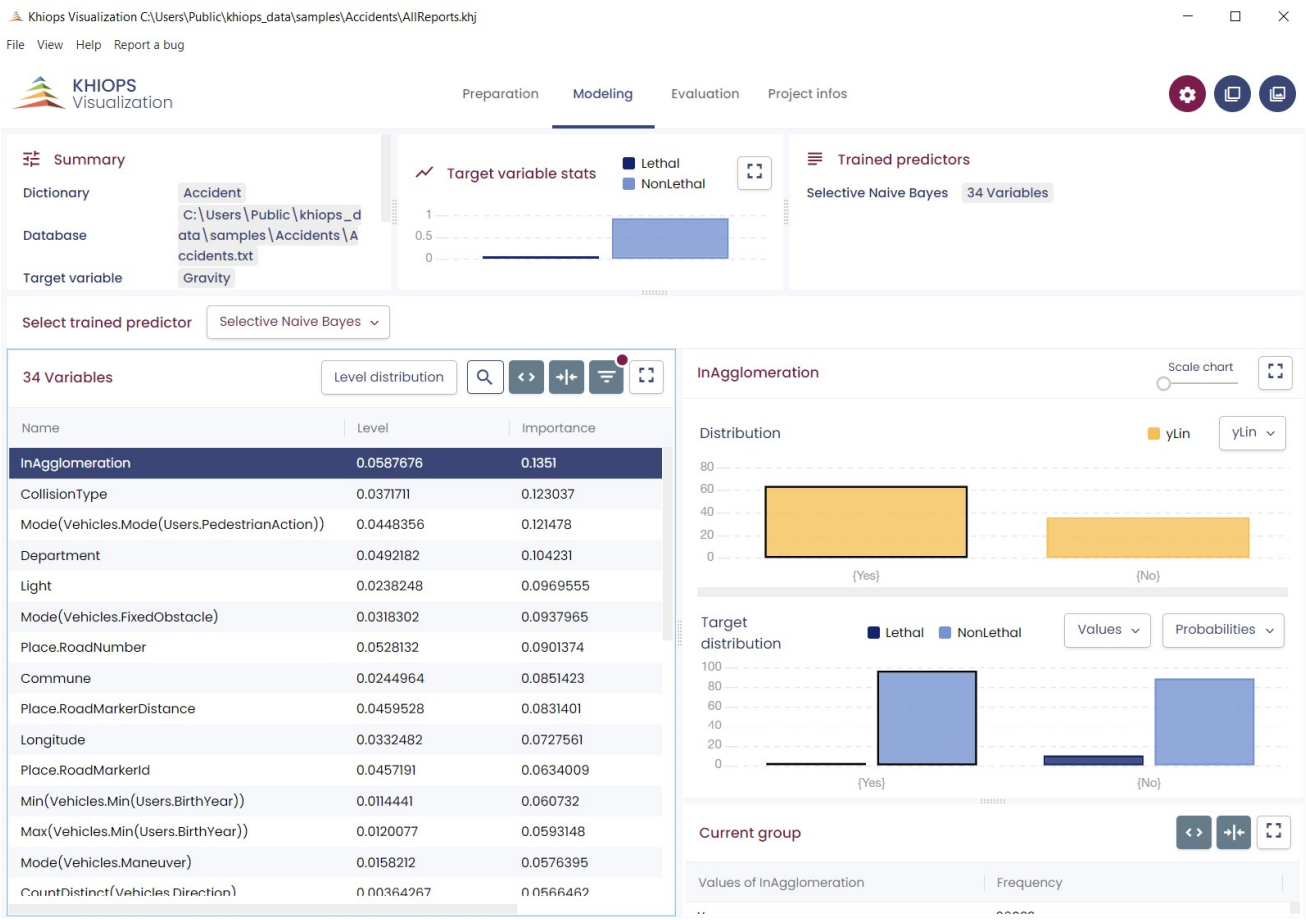


Fig. 4. Screenshot of Khiops Visualisation after analysing the accident database and constructing 100 aggregates

directly, as it can handle categorical and numerical variables. However, for RF and CB we had to preprocess the categorical variables using ordinal encoding. We also report in Table IV the energy consumption in Watt of the three classifiers for their training process, measured using the code carbon library [20] in order to evaluate their energy efficiency [10].

Looking at the results of tables III and IV, we observe several points: the 3 classifiers perform equally well in testing, but Khiops is the most robust (test/train ratio), the least energy-consuming (by a large margin) and the more parsimonious (fewer variables used) which is often desirable to facilitate interpretation.

	Accuracy			
	Train	Test	ratio Test/Train	#variables
Khiops	0,9225	0,9017	0,9774	15
Random Forest	0,9999	0,9005	0,9006	42
CatBoost	0,9871	0,9021	0,9139	41

TABLE III
PERFORMANCES OF THE TREE CLASSIFIERS.

3) Variable importance results: The Figure 5 shows the normalized importance of the variables for each classifier. There are some similarities (example 'ct_srv_dst') but also some marked differences (example 'sttl'). Remember that khiops, being parsimonious, only uses 15 variables. We also give in Table V the first five more important variables for each classifier.

	Energy to train the classifier	
	Energy (W)	ratio Khiops / Competitor
Khiops	2,99 10^{-4}	-
Random Forest	88,73 10^{-4}	30
CatBoost	93,24 10^{-4}	31

TABLE IV
ENERGY CONSUMPTION OF THE TREE CLASSIFIERS

Khiops	Random Forest	CatBoost
sbytes	ct_dst_src_ltm	sttl
sload	ct_state_ttl	ct_dst_src_ltm
sttl	sload	smean
smean	sttl	proto
dbytes	sbytes	sbytes

TABLE V
THE FIRST FIVE MORE IMPORTANT VARIABLES FOR EACH CLASSIFIER

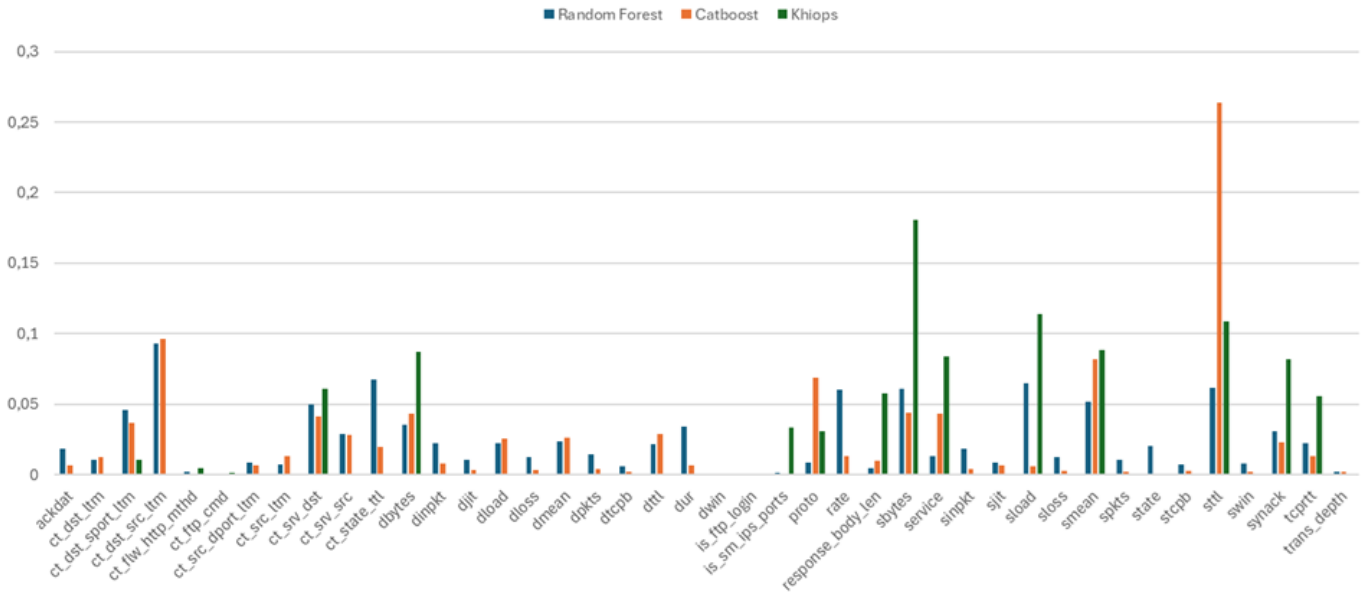


Fig. 5. Variable's Importance for the 3 classifieurs

V. Frugal use of computer resources

In this section, we illustrate how Khiops makes efficient use of computer resources, enabling the tool to analyze datasets that are much larger than the available RAM.

For this experiment, we use the *Zeta* dataset from the Large Scale Learning Challenge [5], which contains $500,000$ training examples and 2000 numerical explicative variables. This is a binary classification problem. This data file takes 9.3 GB on the hard disk, and this run was carried out on a Intel Xeon Gold 6150 CPU 2.70 Ghz.

To illustrate the size of the problem, loading the dataset into memory using Python pandas takes about two minutes and requires approximately 8 GB of RAM. Using an optimized *parquet* data format reduces the loading time by about a factor of 15, but the memory footprint remains the same, not accounting for any additional algorithmic requirements for training a classification model. Analyzing such a dataset is therefore impossible if the available RAM is not significantly larger than the data size.

Using Khiops, the experiment consists in training a classifier and evaluating it, by varying the number of cores and the amount of RAM available. 70% of the examples are used for training and 30% for testing. Figure 6 plots the execution time in minutes, as the number of cores and the amount of RAM increase together. Firstly, the results indicate that Khiops can analyze this large dataset using just 512 MB of RAM and a single core. Due to the limited computational resources, the full processing pipeline takes 81 minutes, whereas it only takes 3 minutes with 32 cores and 16 GB of RAM. Figure 6 shows that there is a smooth transition from out-of-core to distributed computing, demonstrating the efficiency of the adaptation strategy to the available hardware resources. This is made possible by thorough I/O optimization. Finally, you won't be penalized significantly if your hardware is undersized for the task at hand

[5] https://k4all.org/project/large-scale-learning-challenge/

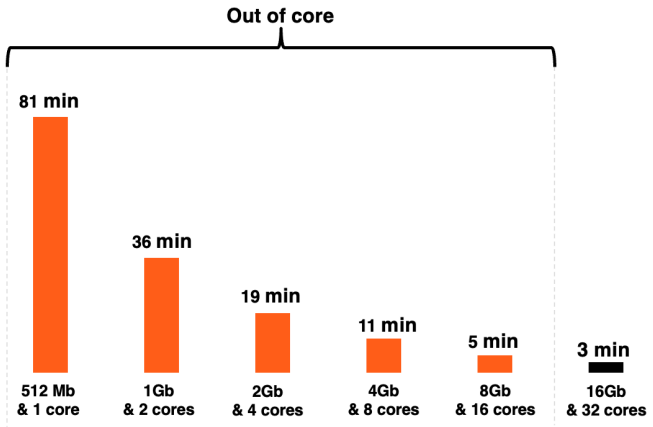


Fig. 6. Calculation time for 9 GB dataset.

VI. Perspectives

Within Orange, major research work is continuing around Khiops, with the release in 2025 of advanced methodologies, such as: robust calibration of classifiers, selection of columns in secondary tables, selection of variables in the presence of concept drift. In the medium term, work will be carried out to process signal-type data (i.e. time series and images) and to develop generative models dedicated to tabular data. More broadly, the MODL approach has been and continues to be studied by the scientific community, with work on association

rules [21], sequence mining [22], clustering [23], [24] , uplift [25] and multi-table variable selection [26], for example.

REFERENCES

[1] A.-M. T. Ehis, "Optimization of security information and event management (siem) infrastructures, and events correlation/regression analysis for optimal cyber security posture," *Archives of Advanced Engineering Science*, pp. 1–10, 2023.

[2] M. Zulfadhilah, Y. Prayudi, and I. Riadi, "Cyber profiling using log analysis and k-means clustering," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 7, pp. 430–435, 2016.

[3] M. Boullé, "Recherche d'une représentation des données efficace pour la fouille des grandes bases de données," Ph.D. dissertation, Télécom ParisTech, 2007.

[4] M. Boullé, "MODL: a Bayes optimal discretization method for continuous attributes," *Machine Learning*, vol. 65, no. 1, pp. 131–165, 2006.

[5] ——, "A Bayes optimal approach for partitioning the values of categorical attributes," *Journal of Machine Learning Research*, vol. 6, pp. 1431–1452, 2005.

[6] N. Voisine, M. Boullé, and C. Hue, "A bayes evaluation criterion for decision trees," *Advances in Knowledge Discovery and Management (AKDM09)*, vol. 292, pp. 21–38, 2009.

[7] M. Boullé, C. Charnay, and N. Lachiche, "A scalable robust and automatic propositionalization approach for bayesian classification of large mixed numerical and categorical data," *Machine Learning*, vol. 108, pp. 229–266, 2019.

[8] C. Hue and M. Boullé, "Fractional naive bayes (fnb): non-convex optimization for a parsimonious weighted selective naive bayes classifier," 2024. [Online]. Available: https://arxiv.org/abs/2409.11100

[9] S. Sonnenburg, V. Franc, E. Yom-Tov, and M. Sebag, "Pascal large scale learning challenge," 2008, http://largescale.first.fraunhofer.de/about/.

[10] L. Arga, F. Bélorgey, A. Braud, R. Carbou, N. Charbonniaud, C. Colomes, L. Delphin-Poulat, D. Excoffier, C. Fauché, T. George, F. Guyard, T. Hassan, Q. Lampin, V. Lemaire, P. Nodet, P. Piotrowski, K. Sapiejewski, E. Sirvent-Hien, and T. Tosic, "Frugal AI: Introduction, Concepts, Development and Open Questions," *SIGKDD Explor. Newsl.*, vol. 27, no. 1, p. 72–111, Jul. 2025. [Online]. Available: https://doi.org/10.1145/3748239.3748247

[11] V. Lemaire, F. Clérot, and M. Boullé, "An efficient shapley value computation for the naive bayes classifier," in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, R. Meo and F. Silvestri, Eds. Cham: Springer Nature Switzerland, 2025, pp. 75–90.

[12] V. Lemaire, N. Le Boudec, V. Guyomard, and F. Fessant, "Viewing the process of generating counterfactuals as a source of knowledge: a new approach for explaining classifiers," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–8.

[13] V. Lemaire, C. Hue, and O. Bernier, "Correlation explorations in a classification model," in *Workshop Data Mining Case Studies and Practice Prize, KDD 2009*, 2009. [Online]. Available: https://www.researchgate.net/publication/377921678_Correlation_Explorations_in_a_Classification_Model

[14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Neural Information Processing Society (NeurIPS)*, 2017.

[15] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[16] A. Bondu and M. Boullé, "A supervised approach for change detection in data streams," in *Proceedings of International Joint Conference on Neural Networks*, 2011, pp. 519–526.

[17] M. Boullé, "Prediction of methane outbreak in coal mines from historical sensor data under distribution drift," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing - 15th International Conference, RSFDGrC 2015*, 2015, pp. 439–451.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 6639–6649.

[19] L. Breiman, "Random forests," vol. 45, no. 1, pp. 5–32.

[20] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, LiamConnell, A. Saboni, Inimaz, supatomic, M. Léval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, N. Laskaris, E. Abati, D. Blank, Z. Wang, A. Catovic, M. Alencon, M. Stęchły, C. Bauer, L. O. N. de Araújo, JPW, and MinervaBooks, "mlco2/codecarbon: v2.4.1," May 2024. [Online]. Available: https://doi.org/10.5281/zenodo.11171501

[21] D. Gay and M. Boullé, "A bayesian approach for classification rule mining in quantitative databases," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 243–259.

[22] E. Egho, D. Gay, M. Boullé, N. Voisine, and F. Clérot, "A user parameter-free approach for mining robust sequential classification rules," *Knowledge and Information Systems*, vol. 52, no. 1, pp. 53–81, 2017.

[23] R. Guigourès, "Utilisation des modèles de co-clustering pour l'analyse exploratoire des données," Ph.D. dissertation, Université Panthéon-Sorbonne-Paris I, 2013.

[24] O. A. Ismaili, "Clustering prédictif décrire et prédire simultanément," Ph.D. dissertation, Université Paris Saclay (COmUE), 2016.

[25] M. Rafla, "A bayesian approach for uplift modeling: application on biased data," Ph.D. dissertation, Normandie Université, 2023.

[26] M. Boullé, "Towards automatic feature construction for supervised classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 181–196.

Real-Time Vessel Identification on Edge Devices

Dominique Heller, Cédric Seguin, Paul Gautier, and Johann Laurent
Université Bretagne-Sud, CNRS-UMR 6285 Lab-STICC, Lorient, France
{firstname}.{lastname}@univ-ubs.fr

***Abstract*—Maritime surveillance plays a critical role in ensuring the security and safety of coastal regions, demanding reliable identification and tracking of vessels across spatially distributed camera networks. This task is particularly challenging as the system must identify a vessel even if it appears in different lighting conditions, angles, or environments. In this work, we propose a real-time visual vessel re-identification system optimized for deployment on edge devices. We evaluate our method on a custom maritime dataset and benchmark its performances on the NVIDIA Jetson Orin NX platform, AMD-Xilinx Kria KV260 Vision AI Kit, and a Raspberry Pi 5 with Hailo-8 accelerator. This work highlights the feasibility of deploying advanced Re-ID systems at the edge, enabling scalable, real-time maritime monitoring solutions. The proposed deployments demonstrate promising results with an inference speed of 20 FPS and a limited degradation of 3% in mean average precision in the worst case due to 8-bit quantization.**

***Index Terms*—Deep-learning, Edge computing, Marine vehicles identification**

I. Introduction

According to the United Nations Conference on Trade and Development (UNCTAD), the maritime transport sector accounts for over 80 percent of world trade volume. It is essential to guarantee the safety and security of maritime traffic from coastguard stations, Maritime Affairs ships patrols, and aircraft. The purpose is to maintain control over all activities related to the maritime environment, including commercial traffic management, sea fishing, and the monitoring of marine pollution, among others. Maritime surveillance is the process of monitoring, detecting, identifying, and tracking vessels and objects in or near a marine environment. It can be conducted using a variety of technologies and methods, including satellite imagery, automatic identification system (AIS), radars and cameras.

Currently, human analysis, assisted by simple intelligent methods, remains the most common approach for processing large-scale maritime surveillance videos. With the decreasing cost of cameras and sensors, the volume of usable data has surged, making analysis increasingly tedious for operators. This time-consuming process is also prone to errors, including missed detections due to operator fatigue. To address these challenges, research on automated identification systems has become a dynamic area of computer vision.

However, maritime environments present unique challenges, including variable weather conditions, the dynamic nature of the ocean, and the vastness of the surveillance area. These factors can decrease the quality of visual data and complicate object detection. Sensor fusion combining visual data such as camera, radar, LiDAR, infrared, or even satellite feeds, allows for better detection, even in low-visibility conditions (fog, night-time, or adverse weather), improving the robustness of maritime surveillance systems.

Additionally, deploying drones with embedded computer vision systems closer to the target area can help overcome challenges posed by the maritime environment, improve identification accuracy, and reduce the cost of maritime surveillance missions. Furthermore, an image-based classification and identification system is complementary to radar and AIS data. It helps to remove doubts about radar detections, which are limited in terms of target classification and identification due to the lack of return information (radar-equivalent surface), and to fill the gaps left by the AIS, such as jamming, shutdown or identity theft.

Re-identifying (Re-ID) vessels is a particularly important task in scenarios such as surveillance or tracking ships over time, across different images or video frames. However, these computer vision algorithms may be too resource-intensive to be embedded on edge devices, such as those on drones. Our research introduces a real-time system for re-identifying marine vessels across different images. This approach that combines a deep learning technique is based on triplet loss network, with embedding analysis, and a k-nearest neighbors algorithm (K-NN). It considers the appearance, category, and orientation of vessels, to determine whether different images represent the same object. The proposed system is designed for deployment on edge computing devices embedded in maritime and aerial vehicles, using initially an RGB camera.

The rest of the paper is organized as follows. Section II presents related work on marine vessel Re-ID, while Section III introduces a new large-scale marine vessel dataset. Section IV provides training analysis, and Section V presents inference results on edge devices. Finally, we discuss our results and future work in Section VI, followed by the conclusion in Section VII.

II. Related works

This section introduces the concept of Re-Identification (Re-ID) by briefly reviewing research on person and vehicle Re-ID, followed by an exploration of recent advancements in marine vessel detection and orientation recognition.

A. Person and Vehicle Re-Identification

Re-ID focuses on retrieving an entity of interest across multiple non-overlapping camera views. This is a challenging computer vision task, as its goal is not only to differentiate

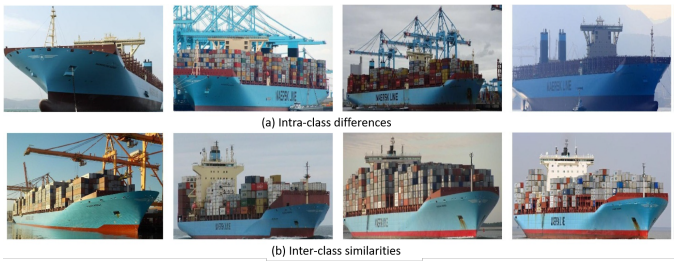


Fig. 1. Complexity of vessel Re-ID [1].

between object categories, as in classification tasks, but also to recognize the same individual objects across different images. The main challenge is to accurately associate the same entity captured under different conditions, including variations in lighting, pose, viewpoint, background, and occlusions. Re-ID is widely used in surveillance through Person Re-ID and in traffic monitoring and automated tolling through vehicle Re-ID.

The main method, in most of the literature of Re-ID, is to locate instances of a query object (probe) from a group of candidates (gallery) captured from different non-overlapping camera views. Features are extracted from each image of a person, and mathematical techniques are used to measure the distance between image pairs.

Building a Re-ID system requires five main steps: raw data collection, bounding box generation, data annotation, model training, and object retrieval.

B. Vessel Re-Identification

Compared with the re-identification of people and vehicles, vessel re-identification offers additional complexities due to several domain-specific factors, such as:

- *Small inter-class similarity*: Vessels from different classes may appear visually similar, especially when they belong to the same ship models or companies.
- *Large intra-class similarity*: The same vessel can look drastically different depending on the viewpoint, making consistent identification difficult.
- *Environmental influences*: Factors such as occlusion, illumination changes, and other environmental noise (fog, rain) further impact the vessel's appearance.

Figure 1, extracted from [1], shows a) the intra-class differences caused by viewpoint changes for the same ship, while b) highlights the inter-class similarity for different vessels of the same type.

Figure 2, shows some sample images of our dataset for different challenging scenarios: different illuminations, variation in scale, change in background, and different viewpoints.

Most of the works consider only similarities and dissimilarities to the vessel identification task, and use the TriNet model [2], or [3]. The loss function optimized to learn such features is the triplet loss. During the learning, it uses three images the anchor (current ship), the positive (another image of the current ship), and the negative (image of another ship). The TriNet

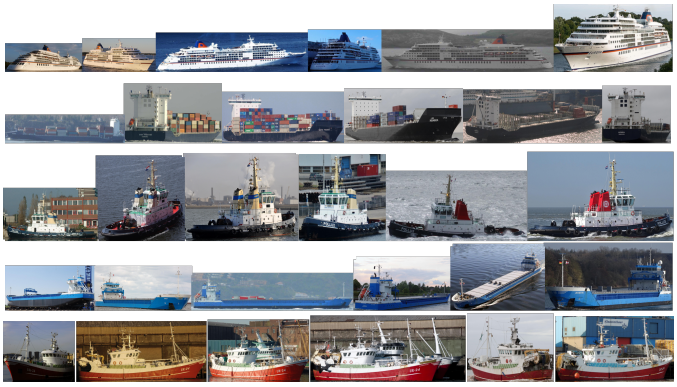


Fig. 2. Samples of our VesselReid-12k dataset. Each row of the figure shows six images of a ship in different scenarios: different Illuminations, variation in scale, change in background, and different viewpoints.

model is trained to minimize the distance between the anchor and the positive sample and to maximize the distance between the anchor and the negative sample. These three images are passed through a Convolution Neural Network (CNN) layer, generating a 1-dimensional feature vectors (embedding) which is used to calculate the distances between them. The CNN is often a customized ResNet-50 architecture that is suitable for embedded applications because of its low computational complexity. At the inference stage, only a single subnetwork is used to generate the embeddings of new input samples and a K-Nearest Neighbors (KNN) algorithm is performed to find the best matches. This approach efficiently identifies vessels based on learned visual features and ensures accurate matching through optimized distance calculations. IORNet [3] proposes an identity-oriented re-identification model that combines triplet loss and cross-entropy loss, using ResNet-50 as the core feature extraction network. [4] study the ship retrieval methods for intelligent water transportation system in smart cities and employed a pyramid structure to deal with variations in ship shapes and sizes. In [5], the authors proposed a quadruplet learning and improve the recognition accuracy taking four images : anchor, positive, negative high-similar (same class vessel) and negative. In [6] the authors proposed a new Vessel Re-ID network (VesselNet), employing ResNet-50 to extract image features and incorporating a hybrid attention module to effectively capture significant features in the images. In [7] a fine-grained feature extraction network (FGFN) is proposed. The authors improve the ResNeSt [8] architecture through incorporating a self-attention mechanism and generalized mean pooling. In [9] the authors propose a two-branch network with dynamic feature enhancement and dual attention to address the issue of low accuracy in ship Re-ID under foggy weather, enabling simultaneous learning of defogging and ship Re-ID tasks in an end-to-end manner.

Specific parts of ships, such as the bow, stern, and equipment on the deck, often possess high uniqueness and discriminability [10]. Capturing these local features allow for more accurate identifications and differentiations of ships, [11] adopts a dual-branch architecture for global and local

feature learning, allowing each branch to focus independently on global or local characteristics.

The GLF-MVFL framework [2] proposes a feature learning method based on global and local fusion, combining cross-entropy loss with orientation-guided quintuplet loss. As large vessels are more sensitive to a change in viewpoint they add two extra image samples (one positive and one negative) to the triplet to constitute the quintuplet : anchor image, a positive image from the same viewpoint, a positive image from a different viewpoint, a negative image from the same viewpoint, and a negative image from a different viewpoint. Compare to IORNet [3], they increased the mean average precision (mAP) and Rank-1 by 7% and 3% with the same backbone ResNet-50 and achieved 74.9% for mAP and 61.4% for Rank-1.

In this article, we propose a large vessel Re-ID dataset called VesselReID-12k and use ResNeSt as the feature extraction network [8]. We also employ generalized mean pooling, hard triplet mining, and re-ranking optimization to achieve state-of-the-art ship re-identification results.

We will describe the large-scale dataset we constructed in the next section.

III. The Large-Scale Marine Vessel Dataset

Since the well-known vessel re-identification datasets VesselReid [12] and VesselReid-539 [2] are not publicly available, we created a large-scale, well-annotated dataset with rich attribute labels, including vessel identities, vessel category (36 classes based on AIS ship types), as well as the main orientations: front, back, one side, oblique front-side, and oblique back-side. The process for collecting, cleaning and annotating data is described in [13].

At the end of this process, our dataset consists of 263,488 images of 12,777 unique vessel IDs, each annotated with a 7-bin orientation and a 36-class vessel-type label.

Figure 2 shows a few representative samples from the dataset VesselReid-12k.

A. Statistical distribution of the dataset

Figure 3 statistically analyzes the VesselReid-12k dataset in terms of vessel types and orientations. This paper classifies vessel types into the thirty-six classes.

It can be noticed that there is an issue of class and orientation imbalance. In future work, we will integrate new images sourced from VesselFinder or Marine Traffic to better balance our dataset. VesselFinder and Marine Traffic are international, free-to-use websites for real-time ship tracking where each boat contains a variable number of images captured from different viewpoints and distances across different times and locations.

In addition, to analyze scale variations within our dataset, as shown in Figure 4 we calculate the mean and standard deviation of the width, the height and the aspect ratio of ship images. The vessel image size and aspect ratio vary greatly. The standard deviation of the aspect ratio (width-to-height ratio) of our dataset is 1.113, which is higher than that found in other Re-ID datasets for ships [11]. Therefore, the greater

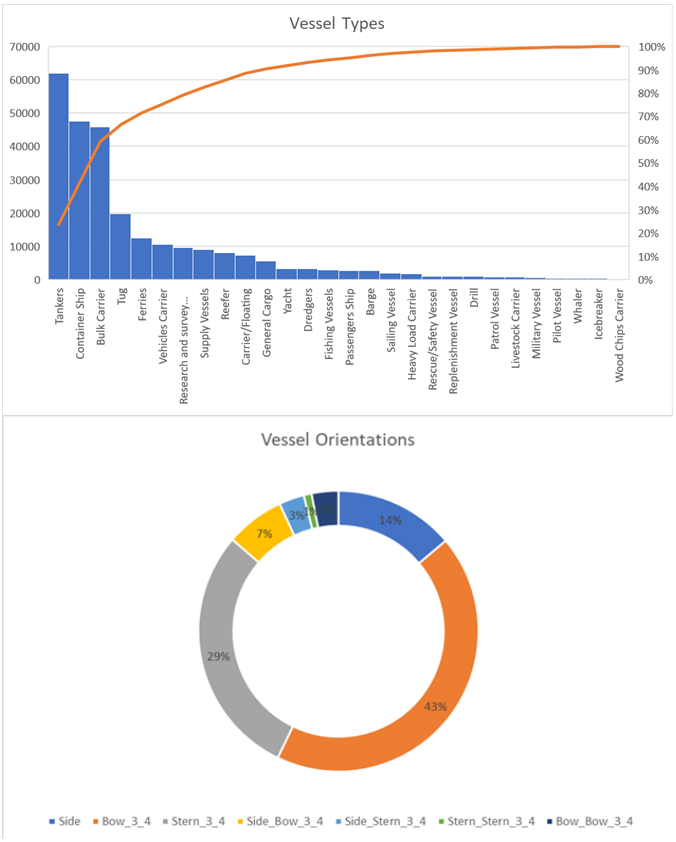


Fig. 3. VesselReid-12k representation in terms of vessel types and vessel orientations.

the diversity of our dataset, the less sensitive the trained model must be to scale change.

B. Comparison with other vessel datasets

Table I presents a comparison between our dataset and other existing vessel datasets. [2] proposed a ship retrieval dataset named VesselID-539, created by selecting images from the Marine Traffic website. The training set contains 104,554 images of 377 identities, while the testing set consists of 44,809 images of 162 identities. [12] introduced a new maritime vessel re-identification dataset named VR-VCA, which includes 729 unique identities along with 5-bin orientation and 8-class vessel-type annotations. [4] constructed a fine-grained ship retrieval dataset (FGSR), consisting of 30,000 field-captured images of 1,000 ships. The VesselID-700 dataset comprises 56,069 images covering seven typical ship classes. Additionally, the Warships-ReID dataset [14] includes 4,780 images of 163 vessels. The ShipReID-2400 dataset is compiled from a real-world intelligent waterway traffic monitoring system. It comprises 17,241 images of 2,400 distinct ship identities collected over 53 months, ensuring diversity and representativeness. Finally, CMShipRied is cross-modality ship re-identification dataset which contains visible light, near-infrared, and thermal infrared modalities collected by autonomous aerial vehicle. It consists of ten categories, about 138 identifications, and 8337 images. Compared to other

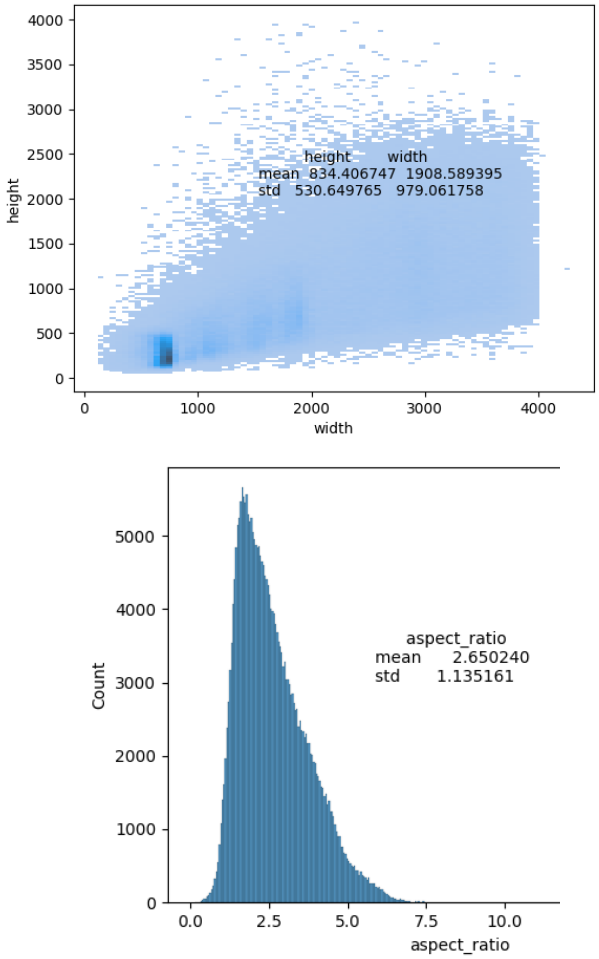


Fig. 4. Scale variation within the VesselReid-12k dataset.

vessel re-identification datasets, our dataset VesselRied-12k contains more ships, more images, and a greater diversity of viewpoints and aspect ratios.

We split our dataset into 3 parts: the training subset includes 7497 IDs and 197616 images, the query (validation) subset contains 5984 IDs with 13175 images, and the gallery (test) subset contains 7481 IDs with 52697 images. In the next section, we will detail the training and associated statistics.

IV. Training Analysis

To distinguish objects based on their physical characteristics at the pixel level, an appearance descriptor called an embedding is generated using a Siamese triplet network. During training, the network uses three images: the anchor (previous detection), the positive (current and future detections of the same object), and the negative (other objects). It is trained to minimize the distance between the anchor and the positive sample, while maximizing the distance between the anchor and the negative sample. These three images are passed through a CNN layer (the backbone), generating a one-dimensional embedding, used to compute the distances. During inference, only one network is used to produce an embedding layer for the candidate track (anchor). Then results are compared to current detection to find the best match based on minimum distance.

To guide feature learning during training, the most commonly used loss functions are identity (cross entropy) loss and triplet loss. Identity loss treats the training process of the identification model as an image classification problem, where each identity ID is considerate as a different class. Triplet loss, on the other hand, considers the training process of the identification model as a retrieval ranking problem, where the distance between positive sample pairs should be smaller than between negative sample one. The choice of positive and negative sample is crucial for improving the retrieval performance of re-identification. This process of selecting effective triplets is often referred to hard mining, while re-ranking is typically applied during the inference stage to refine retrieval results based on the initial ranking.

The basic idea of re-ranking is to utilize gallery-to-gallery similarities to refine the initial ranking list. When fine-tuning with the ranking loss, it is crucial to mine hard triplets efficiently, as randomly selected triplets often result in easy samples or triplets with little, or no loss, contribution.

A. Training process

The training processes are conducted using *FastReid* framework on Nvidia Quadro RTX 3090. The framework FastReID [17] implements state-of-the-art re-identification algorithms. In the image pre-processing stage, resizing and data augmentation techniques such as flipping, cutout, and random erasing are done. The input images are downsampled to resolution of 256×256.

For the backbone that maps images to features representations, four architectures were used and compared: Resnet-50, Resnet50+IBN (with instance batch normalization), Vision Transformer (ViT) and ResNeSt. Transfer learning is adopted by initializing the models with the weight values of models previously trained on popular large-scale ImageNet dataset. The model is trained over 300 epochs.

During training, we use a combination of cross-entropy loss and triplet loss as the loss function. Following the approach in [18], we also apply hard triplet mining to enhance the discriminative power of the triplet loss and mitigate the impact of class imbalance. The aggregation layer is designed to combine the feature maps generated by the backbone into a global feature representation. At this stage, we employ average pooling. For distance metrics, we use the cosine distance, which has yielded better experimental results than the Euclidean distance on normalized embeddings.

B. Evaluation of trained models

The table II shows comparative performance evaluation of 4 backbone configurations. The key performance indicators are the ranked accuracy of re-identification and the mean average precision (mAP). Ranked accuracy is a method of computing accuracy where the top-K highest-confidence predictions are

TABLE I
COMPARISON OF PROPERTIES OF VESSEL RE-ID DATASETS

Dataset	ID Volume	Dataset Scale	Angle of View	Vessel Type
VR-VCA [12]	729	4614	5-bin orientation	8 classes
VesselID-700 [7]	700	56069	5-bin orientation	7 classes
VesselReID [15]	1248	30587	5-bin orientation	7 classes
FGSR [4]	1000	30000	2 cameras view points	none
VeRiS [6]	2,904	150,623	5-bin orientation	7 classes
VesselID-539 [2]	539	149465	superstructure	8 classes
Warships-ReID [2]	169	4780	none	8 classes
VesselReID-1656 [1]	1656	135866	5-bin orientation	12 classes
ShipReID-2400 [11]	2400	17241	8 cameras view points	none
CMShipReID [16]	138	8337	3 image source (VIS,NIR, TIR)	10 classes
Ours (VesselReID-12k)	12777	263488	7-bin orientation	36 classes

TABLE II
TRAINING RESULTS

Model	Feature dimension (Bits)	mAP (%)	Rank-1 accuracy (%)	Rank-5 accuracy (%)	Rank-10 accuracy (%)
Resnet-50	2048	67.05	72.32	89.78	94.89
Resnet-50+ibn	2048	69.66	74.51	90.86	95.60
Vit	768	69.07	73.21	90.89	95.96
ResNeSt	2048	85.19	88.58	96.10	97.90

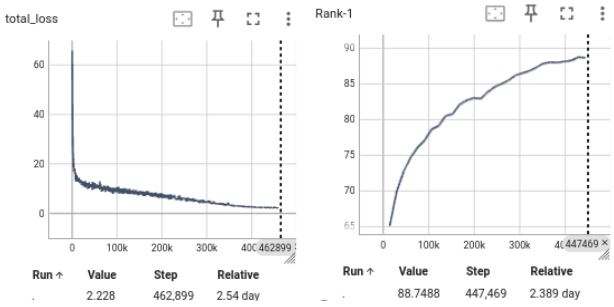


Fig. 5. Training results of ResNeSt configuration: Total (Triplet+Class) loss and rank-1.

compared to the ground truth label. In our case, we compute rank-1 (Figure 5), rank-5 and rank-10 accuracies. This means that for rank-10, if the ground truth label appears among the top-10 predicted labels for a given sample, it is considered as a correct match. The mAP measures the overall prediction accuracy, reflecting how well the model retrieves correct identities. In our case, it evaluates how accurately the model predicts the identity (ID) of a vessel.

The ResNeSt configuration outperforms all other configurations by up to 15% in mean average precision (mAP). It achieves outstanding performance, with a Rank-1 accuracy of 88.58% and an mAP of 85.19%, reaching state-of-the-art levels for vessel Re-ID. In comparison, MVR-net [12] yields a 74.5% mAP and a 77.9% Rank-1 score, while the GLF-MVFL framework [2] achieves 74.9% mAP and 61.4% Rank-1 accuracy.

Training the same ResNeSt configuration on the Market1501 dataset for person Re-ID achieves a Rank-1 accuracy of 95.2% and an mAP of 88.7%. While this performance is slightly better, the complex observation conditions in vessel Re-ID—such as long-range views, foggy skies, and sea reflections—along with the significant variation in vessel sizes and shapes across different viewpoints, reduce re-identification performance compared to pedestrians or vehicles. Additionally, vessels exhibit varying degrees of tilt and different draft depths due to differing loads, further complicating consistent feature extraction.

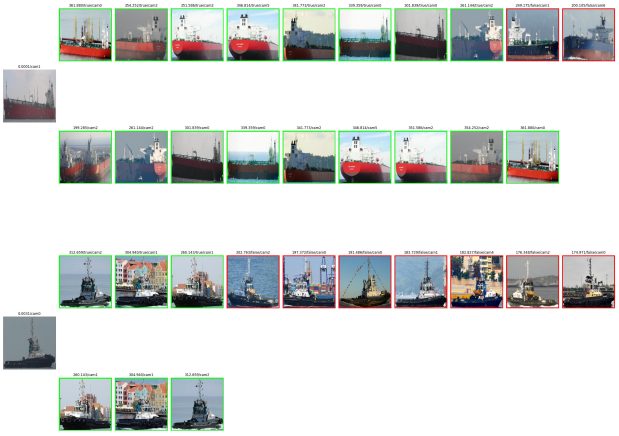


Fig. 6. Illustrations of the top-10 ranking list for retrieval results

We provide representative visualization results to intuitively demonstrate the accuracy of our vessel re-identification model in Figure 6. The left panel shows the query input, while the right panel displays the top-10 retrieved results sorted by similarity. Green boxes indicate correct ID matches, whereas red boxes represent inconsistent re-identification results. In the case of an easy sample (a tanker), the model not only retrieves the correct ID from the gallery set but also ranks it highly in the results. For a hard sample (a tugboat), our model successfully retrieves the correct ID three times, including a correct match at the top-1 rank.

V. Inference on edge device

Deep learning-based object detection on embedded systems must be optimized for low latency, high detection accuracy, and low power consumption. In general, the deployment process comprises two stages. In the first stage, the weights and/or activations are quantized to the desired bit-width and representation (e.g., FP16 or INT8). Quantization refers to the process of converting the weights and activations of a trained deep learning model from high-precision floating-point numbers (e.g., 32-bit) to lower-precision fixed-point or integer representations (e.g., 8-bit). This is typically done using a heuristic method that leverages a selected subset of images from the training dataset, commonly referred to the golden reference pool. This reduction in precision reduces the memory and computation requirements, making it possible to efficiently deploy neural networks on hardware with limited resources.

In the second stage, the quantized model is compiled to generate the instruction sequence. During this stage, the model is further optimized based on the target device's architecture. This study considers three different platforms: one GPU-based, one ASIC-based and one FPGA-based. Their hardware specifications are provided in Table III, which presents the technical details of the embedded edge devices targeted in this work.

A. Deployment targeting Nvidia Jetson Orin Nx

The trained model is deployed using TensorRT to achieve lower latency and higher throughput during inference on NVIDIA platforms. TensorRT is a software development kit (SDK) provided by NVIDIA for high-performance deep learning inference. It is compatible with most deep learning frameworks and is used to achieve high performance and platform portability. It comprises an inference optimizer that implements several techniques, such as kernel fusion, precision calibration, kernel auto-tuning, dynamic tensor memory management, and multi-stream execution, to optimize the inference of the trained model.

Since the *FastReID* framework is not supported by TensorRT, the target model is first converted using the Open Neural Network Exchange (ONNX) format. Next, the model is quantized to FP16 representation. Table IV presents the results in terms of detection performance and inference speed. It compares the original trained model with the TensorRT-converted model when deployed on the Jetson Orin NX. The comparison shows that using TensorRT increases the inference rate, with only a slight degradation in mAP (2.6 %).

TABLE III
Specifications of target edge embedded devices

Target Device	Nvidia Jetson Orin NX 8Go	PI5 + M2 Hailo 8	Kria KV260 Vision AI Kit
Edge accelerator	1024-core 32 Tensor Cores	Hailo 8L	1x DPU configurations B4096 at 300 MHz
AI Performance (estimated FP16)	54 TFLOPS	Unkown	Unkown
AI Performance (estimated INT8)	70 TOPS	13 TOPS	1.43 TOPS
Max Power consumption	30 W	12+3 W	8 W
Price	600$	300$	300$

TABLE IV
Obtained results on Jetson Orin NX

Network	ResNeSt	
Input Resolution	256×256	
Model	Original	FP16
Mean Average Precision	0.852	0.826
FPS	15.7	19.9

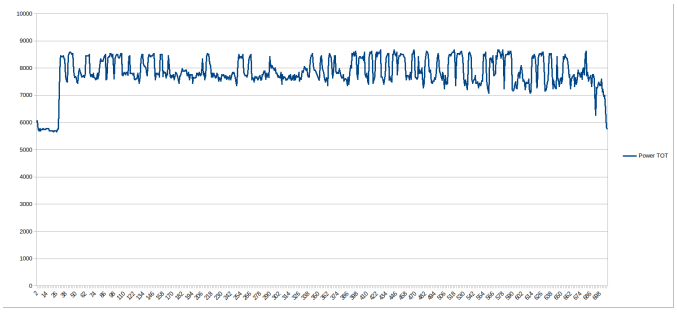


Fig. 7. Power Consumption during inference on query subset on Nvidia Jetson Orin NX.

As the Jetson board incorporates a power monitor, we recorded power consumption during inference on the query subset (5984 IDs with 13175 images) as shown in Figure 7 and average power consumption is around 8 watts for 20 fps, or 2.5 FPS/watt.

B. Deployment targeting RPI5+Hailo 8l NPU

The ResNeSt model is not supported by Hailo's dataflow compiler. Therefore, we used the basic Resnet-50 model for inference. Since one operation (pow 3) is not supported in the aggregation stage either, only the backbone (ResNet-50) was accelerated with the Hailo 8L NPU. The quantization performed by Hailo is a mix of FP16 and INT8 depending on the available resources. After compilation, 60% of the computation resources and 66% of the memory resources of the Hailo8L ASIC are used. The quantization steps results in a 2% loss mAP precision.

The Resnet-50 backbone alone runs at 23 fps. However, with the pre-processing and post-processing code (Head part with GlobalAveragePooling) running on the CPU, the framerate drops to 12 fps. Dispite this, the inference time is sufficient to track ships travelling at typical nautical speeds. Without equipment to measure the power consumption of the Raspberry Pi M.2 HAT, we were unable to determine the power consumption of the Hailo NPU.

C. Deployment targeting Xilinx-AMD Kria KV260 AI Vision Kit

Like the Hailo8 hardware target, only the backbone Resnet-50 model is supported by the framework Vitis AI as clamp and pow operations of GeneralizedMeanPooling layer (Head part) are not supported by the DPU. Also, due to data layout difference between Pytorch training('NCHW') and XIR

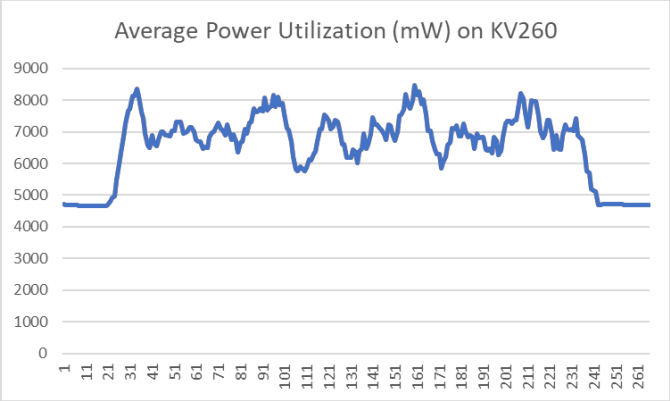


Fig. 8. Average Power utilization (mW) for inference on Xilinx KV260.

DPU('NHWC'), a permution is done on inputs. Using Vitis AI toolset version 3.5, the Resnet-50 model is quantized (Post Training Quantization) into INT8 representation and then compiled targeting DPUCZDX8G architecture. The post training quantization step results in a 3.27% loss mAP precision using the query subset of VesselReid-12K as calibration dataset.

The Resnet-50 backbone alone runs at 24 fps. We recorded power consumption during inference on a subset of the query part (5000 images) as shown in Figure 8 and average power consumption is around 6.8 watts for 24 fps or 3.5 FPS/Watt..

D. Analysis

The Jetson Orin NX is the more powerful device of the three platforms in terms of detection performance (speed and accuracy). It achieves the highest inference speed while maintaining high accuracy with the best vessel Re-ID model, ResNeSt. However, it requires a higher power budget. The Hailo and Xilinx platforms suffer from the limitations of their data flow compilers, which do not support all Re-ID models and require longer development times. Considering performance per watt criterion only, the KRIA KV260 kit outperforms the Jetson Nvidia devices when running Resnet-50 backbone.

VI. Discussions and Future Works

To improve the performance of the appearance descriptor, we plan to build on the approach of [5] and [2] by implementing an enhanced loss function strategy. As depicted in fig 9, we will first introduce the class feature into a quadruplet loss function: an image of another vessel from the same category is added as a second negative sample. This helps to further differentiate intra-class variations. Then, we will develop an orientation-guided quintuplet loss that comprises five images: the anchor image, a positive image from the same viewpoint, another positive image from a different viewpoint, a negative image from the same viewpoint and same category, and a final negative image from a different viewpoint and different category. This loss function is designed to create more robust and discriminative feature representations by considering multiple contextual aspects of the images. To enhance the performance in ship Re-ID in foggy weather, we will study the

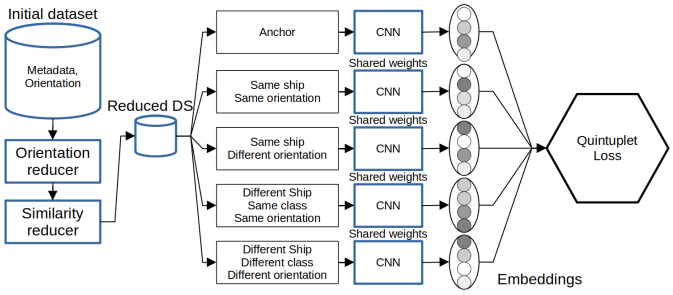


Fig. 9. Training phase

method described in [9] that proposes a two-branch network enabling simultaneous learning of defogging and ship Re-ID tasks in an end-to-end manner. In addition, an atmospheric scattering model [19] is employed for the synthesis of foggy ship images. Quantization usually results in a loss of accuracy due to information lost during the quantization process. For FPGA target, we will use QAT (Quantization-Aware Training) to improve the accuracy of quantization. Finally, Since ships are likely to have distinctive identifying features, such as flags and printed names, our future research will focus on employing a text detector and optical character recognition (OCR). Additionally, as our labeled data have a category field similar to the AIS ship field, we plan to develop a data fusion approach that combines visual detection techniques with Automatic Identification System (AIS) data [20]. Our previous work has already explored real-time classification [21], as well as the fusion of data from cameras, radars, and AIS for classification, situational awareness, and collision avoidance [22].

VII. Conclusion

This paper tackles the topic of Vessel re-identification using deep learning techniques on embedded edge devices. Vessel Re-ID is a challenging task that require to consistently recognize the same boat across different time period contexts and camera viewpoints, despite significant extra-class variations, and intra-class variations caused by changes in viewing angles, illumination conditions, image resolution, occlusions, and appearance alterations such as color shifts. Moreover, maritime environment faces unique challenges, such as changing weather, moving sea conditions, and large monitoring areas. These factors can reduce the quality of visual data and make object detection more difficult. To address the challenges of maritime surveillance, we proposed a real-time solution to vessel Re-ID for maritime surveillance on edge devices. The solution can be carried on drones, equipped with computer vision systems, positioned as close as possible to the target areas. Furthermore, integrating image-based classification and identification with radar and AIS data can enhance the accuracy of vessel identification. We constructed a specialized large-scale dataset for marine vessels, comprising 263,488 images associated with 12,777 unique vessel identities. Each image is annotated with a 7-bin orientation

label and categorized into one of 36 vessel-type classes. This optimized dataset is employed to train a Siamese Triplet Network that learns to generate distinctive high-dimensional embeddings, enabling the computation of similarity distances between entities. During inference, a single network is used to produce an embedding for a candidate track, which is then compared against current detections. The best match corresponds to the detection with the smallest embedding distance. The deployment of the trained models on recent edge devices is considered. We evaluated our solution on a Jetson ORIN Nx, a Raspberry Pi 5 equipped with a Hailo-8 accelerator and the AMD-Xilinx Kria KV260 Vision AI Kit. The results demonstrate that the vessel re-identification system is capable of successfully identifying ships moving at typical maritime speeds with low power consumption. For example, 20 FPS inference speed is achieved on Jetson Orin NX with a mean average precision of 82.6% and average power consumption is around 8 watts. On an embedded system, the choice of hardware target will depend on weight and power consumption constraints. For a system with severe constraints, such as a small flying drone, the best solutions are the Xilinx Kria KV 260 or Hailo-8 accelerator. But for a unmanned surface vehicle, the best solution is the Nvidia Jetson board. It offers greater flexibility, shorter development times, better performance and reasonable power consumption of 8 W.

REFERENCES

[1] Z. Lu, L. Sun, P. Lv, J. Hao, B. Tang, and X. Chen, "A new large-scale dataset for marine vessel re-identification based on swin transformer network in ocean surveillance scenario," *IET Computer Vision*, vol. 19, no. 1, p. e70007, 2025.

[2] D. Qiao, G. Liu, F. Dong, S.-X. Jiang, and L. Dai, "Marine vessel re-identification: A large-scale dataset and global-and-local fusion-based discriminative feature learning," *IEEE Access*, vol. 8, pp. 27 744–27 756, 2020.

[3] A. Ghahremani, Y. Kong, E. Bondarev, and P. H. de With, "Towards parameter-optimized vessel re-identification based on iornet," in *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V 19*. Springer, 2019, pp. 125–136.

[4] Y. Xian, J. Xian, L. Lu, and J. Tang, "Fgsr: A fine-grained ship retrieval dataset and method in smart cities," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 2807139, 2022.

[5] B. Solmaz, E. Gundogdu, V. Yucesoy, A. Koc, and A. A. Alatan, "Fine-grained recognition of maritime vessels and land vehicles by deep feature embedding," *IET Computer Vision*, vol. 12, no. 8, pp. 1121–1132, 2018.

[6] Z. Yu, J. Liu, S. Zou, and Y. Cao, "Vesselnet: A large-scale dataset and efficient mixed attention network for vessel re-identification," in *2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)*. IEEE, 2023, pp. 437–441.

[7] W. Dou, L. Zhu, Y. Wang, and S. Wang, "Research on key technology of ship re-identification based on the usv-uav collaboration," *Drones*, vol. 7, no. 9, p. 590, 2023.

[8] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "Resnest: Split-attention networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2736–2746.

[9] W. Sun, F. Guan, X. Zhang, X. Shen, and K. Wang, "Ship re-identification in foggy weather: A two-branch network with dynamic feature enhancement and dual attention," *Engineering Applications of Artificial Intelligence*, vol. 143, p. 109974, 2025.

[10] M. Zhang, Q. Zhang, R. Song, P. L. Rosin, and W. Zhang, "Ship landmark: An informative ship image annotation and its applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 17 778–17 793, 2024.

[11] B. Liu, R. Huang, X. Pan, C. Li, J. Sun, J. Dong, and X. Wang, "Advancing ship re-identification in the wild: The shipreid-2400 benchmark dataset and d2internet baseline method," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 106–115. [Online]. Available: https://doi.org/10.1145/3726302.3729892

[12] A. Ghahremani, T. Alkanat, E. Bondarev, and P. H. de With, "Maritime vessel re-identification: novel vr-vca dataset and a multi-branch architecture mvr-net," *Machine Vision and Applications*, vol. 32, pp. 1–14, 2021.

[13] C. Seguin, D. Heller, T. Marques, and J. Laurent, "Real-time visual vessel re-identification for maritime surveillance on edge devices," in *OCEANS 2025 Brest*, 2025, pp. 1–7.

[14] G. Zeng, R. Wang, W. Yu, A. Lin, H. Li, and Y. Shang, "A transfer learning-based approach to maritime warships re-identification," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106696, 2023.

[15] Q. Zhang, M. Zhang, J. Liu, X. He, R. Song, and W. Zhang, "Unsupervised maritime vessel re-identification with multi-level contrastive learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5406–5418, 2023.

[16] C. Xu, L. Gao, Y. Liu, Q. Zhang, N. Su, S. Zhang, T. Li, and X. Zheng, "Cmshipreid: A cross-modality ship dataset for the reidentification task," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 10 503–10 513, 2025.

[17] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "Fastreid: A pytorch toolbox for general instance re-identification," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 9664–9667.

[18] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[19] S. Liang, X. Liu, Z. Yang, M. Liu, and Y. Yin, "Offshore ship detection in foggy weather based on improved yolov8," *Journal of Marine Science and Engineering*, vol. 12, no. 9, 2024. [Online]. Available: https://www.mdpi.com/2077-1312/12/9/1641

[20] Y. Lu, H. Ma, E. Smart, B. Vuksanovic, J. Chiverton, S. R. Prabhu, M. Glaister, E. Dunston, and C. Hancock, "Fusion of camera-based vessel detection and ais for maritime surveillance," in *2021 26th International Conference on Automation and Computing (ICAC)*, 2021, pp. 1–6.

[21] D. Heller, M. Rizk, R. Douguet, A. Baghdadi, and J.-P. Diguet, "Marine objects detection using deep learning on embedded edge devices," in *2022 IEEE International Workshop on Rapid System Prototyping (RSP)*. IEEE, 2022, pp. 1–7.

[22] R. Douguet, D. Heller, and J. Laurent, "Multimodal perception for obstacle detection for flying boats-unmanned surface vehicle (usv)," in *OCEANS 2023-Limerick*. IEEE, 2023, pp. 1–8.

Multi-Source and Multi-Target Tracking for Maritime Surveillance

Baptiste Morisse
Lead scientist
Aegir
baptiste.morisse@aegir.fr

Edouard Villain
Research engineer
Aegir
edouard.villain@aegir.fr

Joao Chueire
Research engineer
Aegir
joao.chueire@aegir.fr

Matthieu Vanicat
Chief scientific officer
Aegir
matthieu.vanicat@aegir.fr

***Abstract*—We present a multi-source, multi-target tracking system for maritime surveillance. This system integrates heterogeneous data sources—radar, satellite RF, AIS, EO/IR, sonar, and EW data—differing in nature, update rate, and accuracy. It is designed to process both out-of-sequence measurements or infrequent data, and data with varying levels of identification, from non-existing identifiers to strongly attributed sources. The system operates in real time, even in dense environments with several hundred concurrent tracks. It combines hypothesis management techniques, spatio-temporal data structures, and filters tailored to maritime dynamics. The system's performance is evaluated on synthetic data, and preliminary experiments on real AIS and satellite RF data have shown promising results.**

***Index Terms*—Target tracking; Sensor fusion; Bayesian methods; Maritime navigation; Multisensor systems; Kalman filters;**

I. Introduction

Multi-target tracking consists in performing two main tasks, given a set of partial and noisy measurements from one or more heterogeneous sensors:

- Grouping measurements generated by a same source, often referred to as a target or mobile. Such groups of measures generate a set of tracks, with ideally one track per source. One issue is the presence of sensor false positives (for instance, clutter). A track is a collection of spatio-temporally correlated measurements, carrying potentially an identifier defined at its creation as well as identification or classification information about the source. This step is referred to as *data association*.
- Estimating for each track the position, velocity, and associated uncertainties over time, based on the partial and noisy measurements that compose it. This step is referred to as *data assimilation*.

These two steps are performed jointly and enable the construction of the *Common Operational Picture (COP)*, i.e. the trajectories, current positions and velocities, together with identification and classification information for all mobiles within the sensors environment.

Multi-target tracking has been an active research area for several decades, evolving in response to industrial challenges and the steady increase in available computational power. The earliest industrial needs emerged in the 1960s around airborne radar systems, focusing on the detection and tracking of aircraft and missiles. The first multi-target tracking systems were born from the mathematical formalization of the Kalman filter [1], [2] and probabilistic data association in cluttered environments [3], as well as the advent of microprocessors.

These techniques matured with the introduction of methods such as Probabilistic Data Association (PDA) [4], Joint PDA (JPDA) [5], [6], and Multiple Hypothesis Tracking (MHT) [7]–[9] for robust data association and clutter handling, along with the Extended Kalman Filter (EKF) [10], [11] to handle nonlinear models. These approaches found applications in defense systems, in particular in combat systems and command-and-control (C2) architectures. Over time, their robustness and tracking accuracy were improved through techniques such as the Ensemble Kalman Filter (EnKF) [12], [13], Interacting Multiple Models (IMM) [14]–[16], and Particle Filters and Sequential Monte Carlo Methods (SMC) [17]–[19], enabling the tracking of maneuvering or evasive targets and the use of incomplete or imprecise data.

More recently, a new probabilistic paradigm based on Random Finite Sets (RFS) has emerged for describing the multi-target tracking problem [20]. This makes it possible to describe the tasks of data association and data assimilation in a unified manner within a rigorous and general Bayesian framework, to control the approximations and assumptions made in the modeling, and to state and prove optimality results. It led to algorithms such as PHD [21], [22], CPHD [23] and Multi-Bernouilli [24]–[26] filters, particularly suited to high-noise environments. In parallel, tracking algorithms coupled with raw signal processing [27]–[29] —often leveraging recent advances in machine learning (e.g., computer vision, point cloud analysis [30]–[33])—have been developed to address video, lidar and radar tracking challenges in domains such as Advanced Driver Assistance Systems (ADAS), autonomous driving, and autonomous navigation or decision-making for drones.

This paper addresses the problem of multi-source, multi-target fusion for maritime surveillance. Relevant applications include, but are not limited to: infrastructure monitoring (e.g. ports, offshore platforms), Exclusive Economic Zone (EEZ) surveillance by coast guards, generation of a common operational picture (COP) from onboard ships or drones, or more generally, within a network of distributed sensors and effectors. The system ingests heterogeneous data sources, such as AIS, radar, satellite RF, EO/IR, sonar, and EW data.

The primary challenges lie in handling (1) intermittent, irregular data with highly variable transmission delays, and (2) a large number of simultaneous targets (hundreds to thousands). Data intermittency—such as a vessel becoming undetectable for hours after disabling its AIS—requires the ability to maintain track continuity using large-scale behavioral models. Traditional velocity diffusion models (e.g. random acceleration) perform adequately with high-frequency updates (e.g. every second) but struggle to produce realistic presence probability estimates over longer time horizons (tens of minutes to hours) without measurements.

Highly variable transmission delays—such as satellite data becoming available several hours after acquisition—require the ability to ingest lukewarm data, often arriving out of sequence with respect to their acquisition time, with delays of up to several hours. Lastly, source heterogeneity requires the assimilation of measurements with diverse characteristics and varying levels of kinematic and identification content.

We propose a multi-source, multi-target tracking system that addresses these challenges through several original components:

- A behavioral model for targets, capable of estimating realistic presence probabilities over a time period spanning several hours;
- A spatio-temporal KD-tree data structure enabling efficient retrieval of relevant tracks and time points for assimilating measurements, including out-of-sequence observations;
- Measurement likelihood models that account not only for kinematic parameters but also for various levels of identification features (e.g. RF signatures, MMSI identifiers).

The remainder of the paper is organized as follows. Section II gives an overview of the sensors and data that can be involved in maritime surveillance applications, describes real-world scenarios in which the system has been tested and details the synthetic data generation process used for development and quantitative evaluation of the system. Section III introduces notation sand definitions, and gives a detailed presentation of the algorithmic components. Section IV focuses on results and performance metrics of the tracking system. Finally, section V concludes with a discussion and outlines directions for future works.

II. Sensors and Datasets

A. Sensors and data for maritime surveillance

1) Sensors: A wide range of heterogeneous sensors can be deployed for maritime surveillance applications. Each provides complementary information, differing in coverage, precision, update rate, and robustness to adverse conditions.

- **Global Navigation Satellite Systems (GNSS).** GNSS receivers (such as GPS, Galileo, GLONASS or BeiDou) provide precise latitude, longitude, speed, and heading directly from the vessel instruments themselves. Their advantages include high accuracy (a few meters) and global coverage. However, GNSS is a cooperative signal. It requires the vessel to report its position, and it is vulnerable to spoofing, jamming, or intentional deactivation.
- **Radar.** They provide all-weather, day-and-night detection of vessels, with typical ranges from a few nautical miles (X-band navigation radars) to hundreds of nautical miles (long-range coastal or over-the-horizon radars). Their main characteristics include high update rates (seconds), good range accuracy, and medium angular resolution. However, radars are sensitive to sea clutter and multi-path, and may generate false alarms, particularly near coastlines or in heavy sea states.
- **Automatic Identification System (AIS).** AIS is a co-operative system where vessels broadcast their identity, position, speed, and other navigational information. Its main advantages are the richness of identification data and the high accuracy of reported positions. Limitations include voluntary deactivation, spoofing, delayed transmissions via satellite relay, and incomplete coverage in dense traffic or remote areas.
- **Satellite RF Sensors.** Passive RF payloads on satellites can detect and geolocate maritime transmissions such as AIS, VHF, or radar emissions. They offer wide-area coverage (regional to global) but with low temporal resolution (hours between revisits) Some limitations include high transmission delays and relatively coarse geolocation accuracy compared to terrestrial sensors.
- **Electro-Optical (EO) Cameras.** EO systems provide high-resolution imagery in the visible spectrum, enabling fine-grained classification (ship type, behavior) and situational awareness. They are limited by weather and daylight conditions, and typically offer a narrower field of view compared to radar.
- **Infrared (IR) Sensors.** IR sensors detect thermal emissions, allowing target detection and recognition at night or in reduced-visibility conditions. Their range is generally shorter than radar, and performance is strongly affected by atmospheric conditions (humidity, temperature gradients).
- **Electronic Warfare (EW) Antennas.** EW sensors detect and classify radar or communication signals emitted by vessels. They provide valuable identification cues and operate passively, without revealing the surveillance system. However, their performance depends on the emission behaviors of the target and is then limited in silent or emission-controlled environments.
- **Active Sonars.** Active sonars transmit acoustic pulses and analyze the returned echoes to detect underwater or surface targets. They provide range and bearing measurements with relatively high accuracy and are particularly effective in submarine detection. Main limitations are their limited coverage compared to passive arrays, susceptibility to environmental conditions (e.g. thermoclines), and the fact that emissions can reveal the presence of the surveillance platform.
- **Passive Sonars.** Deployed from fixed buoys, coastal arrays, gliders or naval platforms, passive sonars de-

tect acoustic signatures of vessels and submarines. They provide bearing-only measurements with potential long detection ranges in favorable propagation conditions. Limitations include strong dependence on the underwater environment and difficulty in resolving multiple targets.
- **Human Intelligence (HUMINT).** Reports from coastal patrols, aerial assets, or civilian observers can complement technical sensors. They provide flexible and context-rich information but are irregular in time and highly heterogeneous in reliability.

The core of our tracking system is designed to be sensor-agnostic and be able to operate with data from all of the sensors listed above. To tackle the issue of the variety and heterogeneity of the raw data, and the difficulty therefore to feed them directly to the tracker, we develop and make use of so-called *tactical data*. The idea is to provide a unified blueprint for information required by the tracker, and easily extracted from raw data. Integrating a new sensor therefore only requires specifying its measurement model, i.e. the tactical data it produces and the associated uncertainty. These measurement models are interchangeable components within the tracking system, used both for computing measurement likelihoods and for the filter update step (see Section III).

2) Tactical data: Tactical data are a unified blueprint for processed data, containing target detections and contain (possibly partial) information on position and velocity, as well as identification and classification attributes of the target. An example of tactial data from a raw data would be a raw image (e.g. radar, sonar, or video) that would require preprocessing to extract relevant detections. Following is a short description of such a blueprint:

- Position data: latitude, longitude, range, bearing
- Velocity data: heading, speed, radial velocity
- Classification data: electromagnetic or appearance signature, size, type, track ID, public ID

The capabilities of each sensor are summarized in figure 1.

	GNSS	RADAR	EW	ACTIVE SONAR	PASSIVE SONAR	EO/IR	AIS	SAT RF	HUMINT
Lat/Lon	✓						✓	✓	(✓)
Bearing		✓	✓	✓	✓	✓			(✓)
Range		✓		✓		✓			(✓)
Heading	✓						✓		(✓)
Speed	✓						✓		(✓)
Radial velocity		✓		✓					
Signature			✓			✓		✓	
Size		(✓)			(✓)	(✓)	(✓)		(✓)
Type						(✓)	✓		(✓)
Track Id	✓	(✓)							
Public Id							✓		

Fig. 1. Capabilities provided by heterogeneous sensors for maritime surveillance. Checkmarks indicate typical availability; parenthesized checkmarks denote indirect/conditional availability.

B. Real world experimental cases

The system was tested on real data in two contexts:

- In real time, during the Dronathlon challenge organized by the French Navy. The setup included a USV equipped with a navigation radar, an AIS receiver, and an electro-optical suite, as well as an AUV equipped with a camera. The tracking system fused AIS streams, radar detections, video detections, and navigation information from the drones (MAVLINK). The operational area was limited in size (a few kilometers in radius) and restricted to challenge participants only, resulting in a relatively small number of entities to detect.
- Offline, on historical AIS data and satellite RF detections in the Gulf of Guinea. These data covered an area of several hundred kilometers in radius with several hundred vessels navigating simultaneously. Three satellite passes (one every 24 hours) were available.

These real-data experiments provide only a qualitative evaluation of the system, since omniscient ground-truth information on the operational situation was not available. It is therefore essential to quantitatively assess system performance on synthetic data, for which the ground truth is known. This is the focus of the following subsection.

C. Synthetic data

The generation of synthetic data is essential for the development of a tracking system. It offers several advantages:

- Scenario control: number, type, and trajectories of targets, as well as the type and performance of available sensors.
- Omniscient ground-truth knowledge: enabling rigorous evaluation of tracking system results.
- Mass generation of scenarios: allowing the study of parameter impacts such as noise level, false-positive rate, transmission delays, etc.

To generate synthetic data, we used a proprietary simulator, whose main characteristics are described below. The simulator relies on the Godot 3D game engine, which provides numerous utilities for estimating line-of-sight (ray casting), defining physical behavior laws, and handling 3D terrain models.

1) Simulated sensors:

- **AIS.** AIS messages include position information expressed in latitude and longitude (with Gaussian measurement noise of standard deviation 20 m) together with speed and heading, affected by Gaussian noise with standard deviations of 1 knot and 5°, respectively. Each message also carries the MMSI identifier, which is a persistent and unambiguous identifier of the source. Vessels broadcast AIS messages on average every 5 seconds, with a random transmission delay between 0 and 5 seconds. When a vessel is stationary, the emission rate decreases to one message every 5 minutes. Vessels may probabilistically disable AIS transmissions for random durations of up to 6 hours.
- **Radar.** Radar detections consist of distance (with multiplicative Gaussian noise of 0.5% standard deviation) and

azimuth (with Gaussian noise of 1.5° standard deviation) for all targets within the radar detection range. The radar performs one full revolution every 10 seconds. False negatives (missed detections) occur with a given probability, while false positives (spurious echoes unrelated to any real target) are generated uniformly across the coverage area, also with a given probability per revolution. The radar may attach an ephemeral identifier (tracklet) to detections, modeling the presence of an internal tracking system.

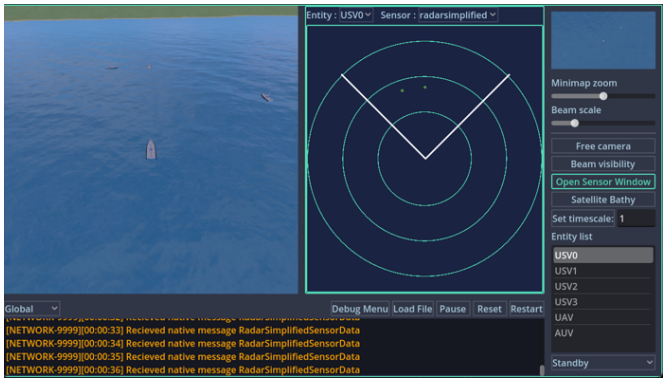


Fig. 2. Simulator user interface, with graphical view of the theater and graphical outputs of a radar sensor

2) Entities and behavior laws: Three categories of entities, each with specific behavior models, are included:

- **Civilian.** Pleasure craft operate in shallow waters (bathymetry between 10 m and 100 m), following random trajectories at speeds between 2 and 25 knots.
- **Fishing vessels.** Fishing boats depart from their home ports, transit to fishing grounds at depths greater than 200 m, and then perform random movements within a restricted one-nautical-mile radius area at speeds between 2 and 6 knots for several hours (fishing operations) before returning to port.
- **Merchant ships.** Merchant vessels perform transits between ports within the theater (or exit points) selected randomly, cruising at a constant speed of 20 knots, except when approaching ports, where they reduce speed.

3) Scenario and test dataset: Using the simulator, we generated one dataset to evaluate the tracking system. The results are presented in Section IV. The test scenario represents the case of some coastal surveillance with a 10 km range radar (without an internal tracker) and AIS data. The theater covers a 70 by 60 nautical-mile area, includes six ports (serving as departure and arrival points for fishing and merchant vessels) At any given tim, around 200 active entities are in navigation. The dataset consists of 4 hours of simulation. In addition to the tactical sensor data, ground-truth positions, velocities, and identifiers of all targets are recorded to enable quantitative evaluation of tracking performance.

III. Methods

A. Definitions and Overall Operation

The main objective of the system is to track entities, that is to make available in real time to the user their position, velocity, and potential metadata (typically, identifiers), and the amount of certainty regarding those intel (quality of data used, noise estimation, precision of the underlying algorithm...). The available information from those entities comes from sensor measurements, which are by nature heterogeneous, potentially delayed, noisy, and so on. The system feed on those measurements and fuse them in a smart way to recover only the needed information.

Our system relies on Targets and Tracks to do so: a *Target* mimics a real entity, via its metadata; and a *Track* represents the kinematic state of a Target. We will see in the following that one Target may have multiple Tracks associated to it at one given time, using *hypotheses*. The design of our system is then twofold:

- **Data Association** – assigns a measurement to a Track, possibly creating a new one.
- **Track Estimation** – estimates a Target's position and velocity, with some confidence scores regarding the estimation.

These two points are highly connected, as Data Association influences the next Track estimations, and track estimation is used to compare Tracks and data.

Data Association is based on a *Multi-Hypothesis Tracking* (MHT) approach. One hypothesis represents a possible interpretation of the observed data, assuming specific associations between measurements and existing Tracks. Multiple hypotheses must be maintained, especially when there is ambiguity about the assignment of a measurement to a Track. The idea is to delay the choice of the "correct" association until future measurements help resolve the ambiguity.

Track Estimation relies on Bayesian inference techniques to estimate the Target state (position, velocity, and associated uncertainties) from partial and noisy measurements. These techniques use a stochastic motion model (i.e. a prior on the Target's dynamics and behavior) along with models for each and every one existing sensors. The *Extended Kalman Filter* (EKF) performs this estimation in the case of Gaussian uncertainties on both motion and measurement noise.

The system maintains a set of Targets, Tracks, and hypotheses. Intuitively, a hypothesis is a proposed description of reality, containing a set of Targets, each with a defined state. A Track is a mathematical object that characterizes a Target's state and ensures its temporal continuity.

Formally:

- Let $\{M_1, M_2, \ldots, M_I\}$ denote the set of Targets.
- Let $\{T_1, T_2, \ldots, T_J\}$ denote the set of Tracks.
- Let $\{H_1, H_2, \ldots, H_K\}$ denote the set of current hypotheses.

Targets form a partition of the Tracks (i.e. a Track belongs to exactly one Target):

$$M_i = \{T_{i1}, T_{i2}, \dots\}, \quad \{T_1, T_2, \dots, T_J\} = \bigcup_i M_i.$$

The Tracks $\{T_{i1}, T_{i2}, \dots\}$ represent the Target M_i in the different description of the reality carried by the hypotheses. A hypothesis is a set of (Target, Track) pairs:

$$H_k = \{(M_{in}, T_{jn})\}$$

where $T_{jn} \in M_{in}$ and all M_{in} are distinct. A hypothesis defines the set of Targets present and the Tracks that represent them. Each hypothesis has a score S_k, interpreted as the log-likelihood of the hypothesis.

The system ingests data *frame-by-frame*, where a *Frame* is a set of measurements $\{z_1, z_2, \dots, z_m\}$ from a single sensor, assumed to originate from different Targets. A Frame is a way to make unity in a world of chaos: where the data comes from several, different, heterogeneous, delayed, noisy sensors, a Frame represents a batch from one sensor and as homogeneous as possible (with respect to time and some potential hyperparameters). This allows for fine-grained, sensor specific and time-wise analysis of a batch of Data, improving greatly the fusion and uses of the data.

Data fusion proceeds through:

1) Preselection of relevant Tracks
2) Estimation of Track positions at the measurement time
3) Computation of association likelihoods
4) Determination of the best associations
5) Update of hypotheses and Tracks

B. Detailed processing

1) Gating and Preselection of Relevant Tracks: Incoming measurements allow the system to preselect relevant Tracks, that is the most likely to be associated with the measurements. Tracks that cannot explain any measurement with sufficient certainty are discarded. This greatly reduces the computational complexity of the following steps.

This selection is done by computing distances between measurements and Tracks, keeping only those within a gating threshold derived from the sensor and motion models. To perform this efficiently, a KD-tree data structure is used to retrieve the nearest Tracks in logarithmic complexity.

2) Motion Model and Position Estimation: Each Track estimates the Target's position at any time using a motion model and Bayesian filtering (EKF).

We introduce an original motion model adapted to maritime dynamics over long time horizons: A Target has a probability $\lambda \cdot dt$ of changing heading and $\mu \cdot dt$ of changing speed during a time interval dt. As $dt \to 0$, the number of heading and speed changes follows Poisson distributions with parameters λ and μ respectively. When a heading change occurs, the increment is drawn uniformly from $[-\pi/2, \pi/2]$. When a speed change occurs, the new speed is drawn uniformly from $[v_{\min}, v_{\max}]$.

The first and second moments of the state vector $(x(t), y(t), v(t), \theta(t))$—where $x(t)$ and $y(t)$ are Cartesian coordinates, $v(t)$ the speed, and $\theta(t)$ the heading—can be computed analytically, yielding a Gaussian approximation of the stochastic dynamics, compatible with an EKF.

3) Likelihood Computation: Measurement-to-track associations are based on an association cost combining kinematic proximity and identification information.

Kinematic proximity is computed from the predicted Track state at the measurement time, with the sensor model providing the log-likelihood.

Identification information is used to refine this score. Identifiers may be:

- Strong/unambiguous: MMSI, radar track ID, combat system track ID, ...
- Partial: RF signature, visual features, ...
- Absent: basic radar, ...

Identifiers may be persistent or change over time. The diversity of cases justifies the need for sensor-specific exploitation strategies and the careful design of Frames.

4) Association: For each hypothesis H_k, an association cost matrix C_{ij} is built, where i indexes measurements and j indexes Tracks in H_k.

From this matrix, the best global association functions f are computed, with $f(i)$ giving the Track index for measurement i, ensuring $f(i) \neq f(i')$ for $i \neq i'$. The best associations minimize:

$$L_f = \sum_i C_{i,f(i)}.$$

They are computed using the Jonker-Volgenant (JV) algorithm and Murty's algorithm.

5) Update: The best associations for each of the K hypotheses are combined to form the new K best hypotheses. Specifically, we determine the K pairs (k, f) minimizing $S_k + L_f$, where k is the current hypothesis index and f is a global association for that hypothesis.

These pairs define the new hypotheses and update the Tracks. For (k, f), let $H_k = \{(M_{in}, T_{jn})\}$ be the Target-Track content of hypothesis k. All Tracks T_{jn} are updated using the EKF update step with the measurement assigned to j_n by f.

C. Cleaning up and Outputs

As we said at the beginning of this Section, the goal of the Tracker is to output the states of the tracked entities. The core algorithms of our Tracker produce many Tracks and Targets, often more than the "real" number of entities. To maintain those numbers in check and not overwhelm the end user, our Tracker has several way to clean up Tracks:

- Confirmation: a Track is said to be confirmed if it contains some signed data, or enough unsigned data ; an unconfirmed Track is discarded.
- Activation: a Track not receiving data for more than some threshold defined in advance is put to sleep. This avoids Track with too much uncertainty on their state, which could attract too many sound data from other entities.

IV. Results

Simulated data and the tracker described in Sections II and III were used to obtain the following results. A Tracker such as the one designed and studied here may be analysed and studied via various methods. We focus here on two of them: classification tools and trajectories evaluation.

In Subsection IV-A we study our Tracker as a classifier regarding the Data Association process. One advantage using simulated data over real data is the knowledge of the ground truth, via the GPS and medatadata of each entity. As a Track is associated with only one Target, which is defined by a unique set of metadata (in the present situation: AIS metadata), we can pass on the "truth" from the the data to the Tracks themselves. **A Track is said to be the entity j if it contains AIS data from the entity j.** We can then say if a radar data has been correctly associated to the right Track or not.

In Subsection IV-B we study our Tracker through Tracks as whole: does one Track reproduces with high precision the path of the underlying entity? To simplify here the analysis, we compare only Tracks associated with signed (AIS) data, meaning, we already know which entity to compare to.

We partition all Tracks produced by the Tracker as follow, using in particular the confirmed/unconfirmed distinction explained in the previous Section:

- **Signed Track:** is confirmed and contains signed (AIS) data.
- **Ghost Track:** is confirmed but lacks identifier.
- **Unconfirmed Track:** is unconfirmed.

In a real situation, a Ghost Track could be for instance a jet-ski, a drone, or any object without identification (as a malicous or not intent). In our dataset, all entites are signed thanks to AIS metadata, hence no Ghost Tracks should exist if our Tracker were perfect.

A Track is considered pure relatively to one Sensor if all data from the Sensor associated to the Track comes from only one entity.

A. Data Association analysis

We focus here on the quality of the radar data associations. The AIS data association is automatically perfect, as we assume here a high confidence on the AIS metadata (i.e. no spoofing or other techniques).

Radar data are twofold: echoes from actual entities, and false positive like echoes from birds, coastlines or waves. In our dataset, there are 26131 echoes from entities, and 11 false positive. As explained in the previous Section, false positive of the radar should directly be associated with an unconfirmed Track. Finally, a radar data from entity j should be associated to the Track j ; several type of errors may occur: association to another signed Track, or a Ghost Track, or an Unconfirmed Track. Table I sums up the discussion.

Table II presents the distribution of signed, unconfirmed, and ghost tracks, along with other statistics.

	Signed j	Signed k	Ghost	Unconfirmed
radar from j	True	False	False	False
false positive	False	False	False	True

TABLE I
Truthness table

	Signed	Ghost	Unconfirmed
Count of Tracks	601	57	18
Count of pure Tracks	532	49	16
# Radar	25607	511	24
# correct Radar	25397	0	11

TABLE II
Details of tracks

The Table I allows to compute for instance the (global) accuracy of our Tracker regarding radar data association:

$$acc = \frac{25397 + 0 + 11}{25607 + 511 + 24} = 97.2\%$$

Though the accuracy is a very classic metric for classifiers, in our case a more detailed analysis is necessary to understand how the Tracker fails.

Signed Tracks comprise a total of 601 tracks, of which 88.5% of are pure, and contained most of the radar data (98% total). The accuracy is as high as 99%, meaning radar data associated to Signed Tracks are almost always associated to the right Signed Track. Some entities generated multiple tracks, typically for entities remaining in ports for some time. This is a byproduct of the deactivation of Tracks as explained in the previous Section (see Figures 3 and 4 for an example of an entity tracked by two tracks).

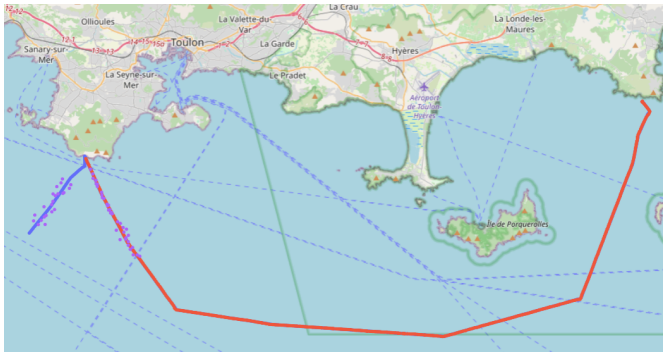


Fig. 3. First part of an entities tracked by two tracks (with red : tracker output; purple : radar input and blue : GPS ground truth)

Unconfirmed Tracks account for 18 tracks in total and less than 0.1% of the radar data. Eleven of these correspond to false-positive radar data points, resulting in 11 pure, unconfirmed Tracks, meaning all unsound radar data have been correctly discarded as unconfirmed Tracks.

Ghost Tracks amount to 57 in total, comprising 511 radar data points (1.9% of all radar data). Among these, 49 were classified as pure, representing 424 radar data points, while the remaining eight accounted for 87 points. Of particular note, 498 out of these 511 radar data points originated from the

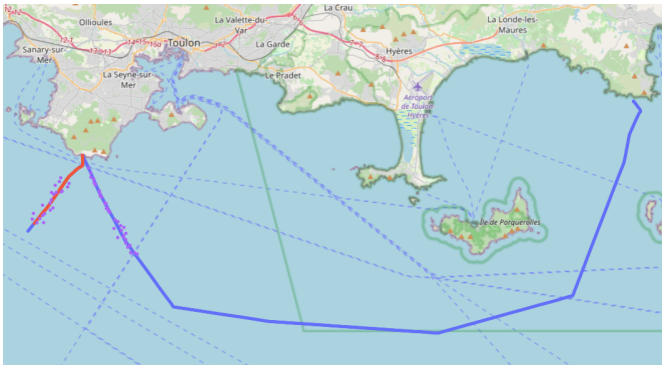


Fig. 4. Second part of an entities tracked by two tracks (with red : tracker output; purple : radar input and blue : GPS ground truth)

three entities that had deactivated their AIS while in motion, thereby simulating malicious behaviour; however, they were still detected by the radar sensor during this AIS interruption. Indeed these ghost tracks contain unsigned data and are used to raise an alert for the end user.

Finally, the tracker was run for 45 minutes in order to assimilate four hours worth of data (see II-C for input data details).

B. Trajectories analysis

As a complement to the previous analysis focused on Data Association, this Subsection focuses on the study of trajectories, comparing Signed Tracks to the path of the corresponding entities.

To this end we use Trajectopy, a python module dedicated to trajectories comparison [34]. A spatio-temporal matching procedure is apply to select the GPS points of the entity's path that best correspond to the associated Signed Track trajectory. The Absolute Trajectory Error (ATE) is then computed, providing metrics that describe the similarity between the predicted trajectory and the ground truth. The three most relevant ATE metrics are used : the minimum, median, and maximum position deviation between the tracker output and the ground truth, in meters. Table III presents the minimum, median and maximum values of Signed Tracks for the three ATE metrics.

ATE Metrics	Minimum	median	Maximum
Minimum position deviation [m]	0.05	1.0	48.8
Median position deviation [m]	8.9	30.6	65.3
Maximum position deviation [m]	25.1	116.8	3811.9

TABLE III
ABSOLUTE TRAJECTORY ERROR SCORES ACHIEVED ON SIGNED TRACKS (IN METERS)

Figure 5 complements Table III by detailing the distribution of the median position deviation, in metres, for signed tracks. It is of particular importance that most of the signed tracks exhibit a median position deviation of less than 30 metres.

Based on the aforementioned ATE metrics and scores, a few tracks were selected. Figure 6 presents one of the best

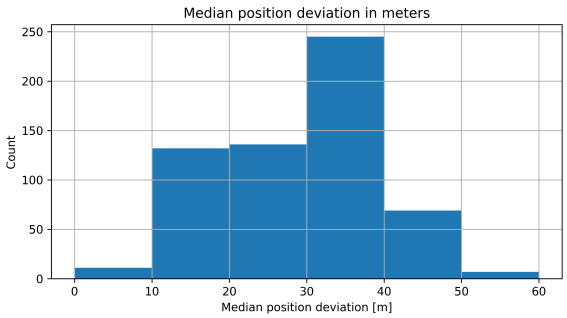


Fig. 5. Median position deviation histogram between tracker prediction and ground truth

signed tracks in this study, with 100% accuracy on radar data. The ATE metrics for this track show minimum and maximum position deviations of 0.21 and 67.7 meters, respectively, while the median position deviation is less than 11.2 meters.

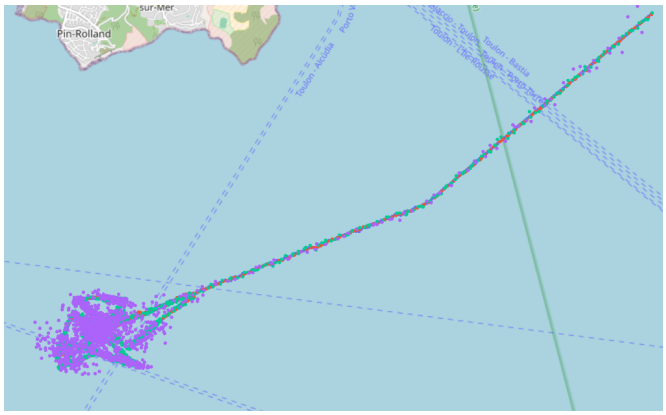


Fig. 6. Track with 100 % accuracies on AIS and radar data (with red : tracker output; green : AIS input; purple : radar input and blue : GPS ground truth)

Figure 7 presents one of the worst signed track in this study, with a data point 3,811 meters from the ground truth. It is worth noting, however, that this track accounts for only 3 errors in radar data, alongside 232 AIS and 99 radar good associations.

V. DISCUSSIONS

This work paves the way towards a generic system for a Common Operational Picture (COP) in the maritime domain. It is designed to integrate any sensor delivering tactical data (that is, kinematic information, optionally enriched with identification cues) and to scale up to complex, high-density scenarios involving hundreds or thousands of tracks. Detailed performance results are provided on synthetic data in a coastal surveillance scenario involving both radar and AIS streams. The system has also been tested on real data in two contexts:

- in real time, during the *Dronathlon* challenge organized by the French Navy, by fusing AIS streams with video detections and navigation information from friendly drones (MAVLINK);

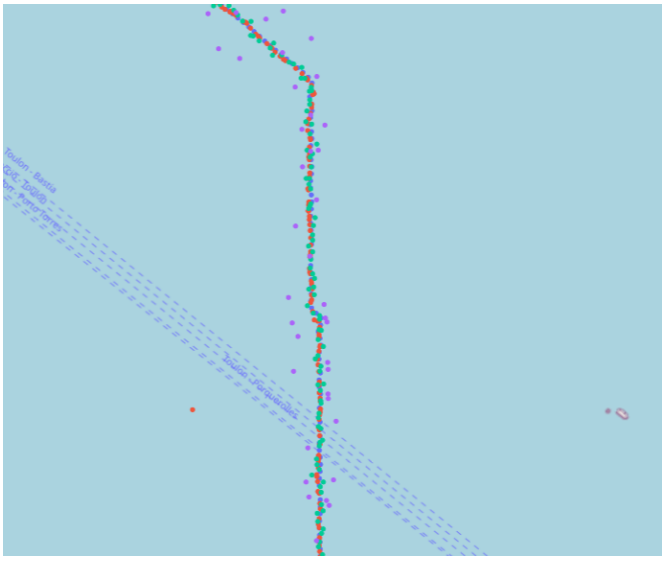


Fig. 7. Track with the highest maximum position deviation (3811 m) (with red : tracker output; green : AIS input; purple : radar input and blue : GPS ground truth)

- offline, on historical AIS data and satellite RF detections in the Gulf of Guinea.

However, several directions remain open for further development.

1) On the theoretical side: There are several elements which may enhance robustness, performance, and applicability of the current system:

- Incorporating ensemble Kalman filters could improve track initialization robustness, particularly in bearing-only scenarios (e.g. passive sonar or EW antennas).
- Leveraging Interacting Multiple Models (IMM) would allow for better tracking of highly maneuverable targets (e.g. USVs, jet-skis), especially in infrastructure protection or onboard COP applications - situations where both spatial and temporal scales are smaller.
- Developing more realistic sensor models and more robust track initiation strategies would help mitigate the impact of spatio-temporally consistent false detections (e.g. false echoes generated by ship wakes);
- Accounting for extended targets that generate multiple detections on a sensor's frame.

2) On the implementation side: Several performance bottlenecks could benefit from parallelization, including filter prediction and update steps, as well as the computation of optimal data association hypotheses. The latter is more delicate and could be addressed by decomposing the bipartite measurement-to-track association graph into connected components for parallel processing, or by parallelizing the partitions evaluated in Murty's algorithm.

3) From an application perspective: Adding an intelligent analysis layer on top of the tracking outputs would enable several valuable functionalities, such as:

- Automatically raising alerts when a vessel disables its AIS, or when a detection cannot be correlated with any AIS message.
- Detecting data inconsistencies (e.g. spoofing attempts, sensor biases).
- Computing metrics for each sensor, such as effective coverage area, consistency index relative to other sensors, update frequency, and transmission latency.
- Computing metrics for each track, such as classification uncertainty, historical richness, regularity, and diversity of contributing sources.

4) In terms of testing and qualification: The system needs to be evaluated more extensively on both synthetic and real data. For synthetic data, it is necessary to:

- Implement within the simulator all sensors listed in the first paragraph of section II.
- Integrate "bearing-only" sensors into the scenario, such as EW antennas or passive sonars.
- Include satellite sensors with transmission delays of several hours, such as satellite RF and satellite imagery.
- Study the tracker's sensitivity to radar false-positive rates and to AIS outages.

It would be beneficial to expand the evaluation tools and performance metrics in order to analyze the tracker's behavior in greater detail and to facilitate its tuning across different application contexts. It is also important to experiment with and validate the system on real data, involving a wide variety of sensors and data sources:

- Radar sensors and EO/IR streams, which may produce large numbers of false positives, sometimes with spatio-temporally correlated distributions (e.g. ship wakes, echoes from obstacles or coastlines).
- EW and acoustic data.
- Fusion of tracks from a combat system or from a tactical data link network (Link 16 or Link 22).
- Integration of human intelligence reports.

ACKNOWLEDGMENTS

The authors would like to thank the French Navy for the opportunity to participate in the Dronathlon, and Unseenlabs for providing satellite RF data.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[3] D. L. Snyder, "Filtering and detection for doubly stochastic poisson processes," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 91–102, 1972.

[4] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[5] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[6] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Applications and Advances. Volume I.* Artech House, 1990.

[7] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[8] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[9] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[10] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[11] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[12] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, 1994.

[13] ——, "The ensemble kalman filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[14] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[15] X. R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure—part i: Model algorithms and part ii: Mode probability computation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 2, pp. 448–468, 1996.

[16] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley-Interscience, 2001.

[17] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[18] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[19] A. Doucet, N. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[20] R. P. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

[21] ——, "Multi-target bayes filtering via first-order multi-target moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[22] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[23] R. P. Mahler, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1523–1543, 2007.

[24] B.-N. Vo and W.-K. Ma, "The multiple target member filter: A new approach to multi-target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1226–1245, 2005.

[25] B.-N. Vo, B.-T. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-bernoulli filter and its implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.

[26] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the bayes multi-target tracking filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6554–6567, 2014.

[27] W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 2, pp. 880–886, 1999.

[28] R. P. Mahler, "Statistics 101 for data fusion: Tracking and detection before tracking," *Proceedings of the 12th International Conference on Information Fusion*, pp. 1409–1416, 2009.

[29] B.-N. Vo, B. Ristic, and B.-T. Vo, "A particle method for multi-target detection and tracking from image observations," in *13th International Conference on Information Fusion*, 2010, pp. 1–8.

[30] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[31] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.

[32] X. Weng and K. Kitani, "A baseline for 3d multi-object tracking," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 20–35.

[33] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 784–11 793.

[34] G. Tombrink, A. Dreier, L. Klingbeil, and H. Kuhlmann, "Spatio-temporal trajectory alignment for trajectory evaluation," *Journal of Applied Geodesy*, 2024. [Online]. Available: https://doi.org/10.1515/jag-2024-0040

WarNav: An Autonomous Driving Benchmark for Segmentation of Navigable Zones in War Scenes

Marc-Emmanuel Coupvent des Graviers[1], Hejer Ammar[2], Christophe Guettier[1], Yann Dumortier[1], Romaric Audigier[2]

[1] Safran Electronics and Defense, Massy, France
{marc-emmanuel.des-graviers, christophe.guettier, yann.dumortier}@safrangroup.com

[2] Université Paris-Saclay, CEA, List, F-91120 Palaiseau, France
{hejer.ammar, romaric.audigier}@cea.fr

***Abstract*—We introduce *WarNav*, a novel real-world dataset constructed from images of the open-source DATTALION repository, specifically tailored to enable the development and benchmarking of semantic segmentation models for autonomous ground vehicle navigation in unstructured, conflict-affected environments. This dataset addresses a critical gap between conventional urban driving resources and the unique operational scenarios encountered by unmanned systems in hazardous and damaged war-zones. We detail the methodological challenges encountered, ranging from data heterogeneity to ethical considerations, providing guidance for future efforts that target extreme operational contexts. To establish performance references, we report baseline results on *WarNav* using several state-of-the-art semantic segmentation models trained on structured urban scenes. We further analyse the impact of training data environments and propose a first step towards effective navigability in challenging environments with the constraint of having no annotation of the targeted images. Our goal is to foster impactful research that enhances the robustness and safety of autonomous vehicles in high-risk scenarios while being frugal in annotated data.**

***Index Terms*—Dataset - Annotation - Semantic Segmentation - Unstructured Environments - Navigability - Data frugality.**

I. Introduction

Modern warfare presents significant challenges for the tactical mobility of mounted combat vehicles. Due to contested environments (GPS-denied, RF-denied), vehicles such as battle tanks, infantry fighting vehicles, and autonomous robots cannot rely on outdated operational pictures to achieve mission objectives. Intensive indirect fire rapidly alters navigable space and key-terrain positions, affecting mission feasibility. Tactical missions now require tight integration of situation awareness and *just-in-time* planning. Furthermore, dominant threats (e.g., loitering ammunitions, short loops between UAV and artillery, remote navigation of drones and robots, or improvised explosive devices) further limit navigable space.

These challenges, rooted in the dynamic nature of the battlefield and the diversity of threats, reveal critical limitations in current mobility and navigation systems. While autonomous navigation technologies in modern urban scenes have been widely developed with rich perception modules owing to finely annotated semantic segmentation datasets, their applicability in hostile, unstructured, and destructed combat zones remains highly constrained. In fact, in these situations, robot autonomy or driver assistance will require strong advancements to navigate efficiently in the no man's land. Moreover, due to the lack of geometrically structured shapes, the differences between two scenes are difficult to assess, even by a human expert, and limited dataset is available to reasonably master the learning bias.

A partial workaround to data scarcity consists of leveraging publicly available information, through techniques such as web scraping, to gain additional information on the target environment. However, the incorporation of extra-military data introduces additional risks [1]. In particular, publicly accessible sources may be subject to intentional manipulation, including large-scale image tampering or disinformation campaigns [2].

In this paper, we propose *WarNav*, a war-zone-specific dataset constructed from the DATTALION repository [3] to support the development and evaluation of robust semantic segmentation models for navigability purposes in conflict-affected settings. The goal is to bridge the domain gap between traditional urban driving datasets and the operational realities faced by unmanned systems in hazardous areas. The central challenges lie in collecting, filtering, annotating, and validating imagery that is both representative and ethically sourced, while establishing procedures that ensure the resulting dataset meets the rigorous standards required for both academic research and practical deployment. Several techniques have been applied to meet these criteria for *WarNav*. Indeed, semantic class labels tailored to navigation tasks are proposed for the test and validation sets to enable performance evaluation.

Moreover, we report baseline performances of several models trained on available annotated datasets without any exposure to *WarNav* images. Test and validation sets are used to evaluate them in war-zone challenging regions, by varying the model architectures, the backbones, and the memory footprints. We also assess the impact of training data domains, ranging from urban to rural and from structured to less-structured environments, on segmentation effectiveness. Results highlight that each domain offers unique benefits towards robust navigability in destructed outdoor areas. Finally, we propose a simple yet effective frugal approach that delivers strong perception capabilities under resource constraints.

Our contributions can be summarized as follows:

- We introduce a novel and challenging use case for semantic segmentation in war-damaged environments, targeting frugal autonomous navigation.
- We construct the *WarNav* dataset via a pipeline of image selection, filtering, curation, and annotation, with a strong focus on ethical sourcing, providing practical insights for future dataset design in extreme deployment scenarios.
- We provide performance on *WarNav* of diverse baselines by varying models, backbones or training environments, and propose an initial frugal approach achieving effective navigability segmentation in conflict-affected areas.

II. Autonomous Robot Use case presentation

A. Goal and challenges

The advance of autonomous and assisted driving technologies is highly dependent on the availability of extensive, high-quality datasets for model development and validation. However, most of the existing datasets for semantic segmentation in the context of ground vehicles, such as Cityscapes [4] or KITTI [5], are predominantly collected in highly structured and undisturbed urban environments. This limits their relevance and utility when models are deployed in more complex, degraded, or unstructured real-world contexts. Through this use case, our aim is to contribute not only with a valuable data resource for the research community but also methodological guidance for future efforts in dataset construction for extreme or atypical operational contexts.

B. Semantic Segmentation of Navigable Spaces

One particularly challenging use case arises in the domain of military operations, where unmanned ground vehicles (UGVs) are expected to perform autonomous navigation tasks in environments characterised by significant destruction, involving debris, destructed vehicles, shell holes, ruts, collapse of buildings, or landslides. In such contexts, accurate perception is critical for both navigation effectiveness and safety. Specifically, the characterization of drivable areas with obstacles can be improved using semantic segmentation. Thanks to semantic retrievals, on-board planners can provide navigation instructions (maneuvers, paths, trajectories) for automatic path and mission completion. However, data scarcity is a major limitation: operational constraints and safety concerns make it impractical to acquire and exhaustively annotate large-scale, representative image datasets in these environments.

C. Frugality needs for autonomous navigation with local situation awareness

Autonomous driving in complex, destructured or unstructured environment must be robust to changes. In particular for ground robotic, mission planning and execution must account for the ability of the autonomous system to interpret its environment, using semantic segmentation among other mission information available on board [6]. Moreover, typical deployment of robotics in military context implies late in-situ image acquisition. It thus can rely on model adaptation during mission preparation [7] through three main phases:

- At mission preparation time, where rough data terrain are available, but not necessarily representative of the battlespace environment.
- After the first mission execution, where some sparse data are gathered from the executed navigation plan. This would correspond to a first major model adaptation.
- During repetitive mission operations, where incremental model adaptations could be performed thanks to incremental data retrieval.

D. Providing dataset from conflict zones

To address this challenge, we turn to publicly available resources that offer authentic, situationally relevant visual content. The DATTALION repository [3] is a prominent example, providing visual documentation from Ukrainian conflict zones, reflecting the diversity and chaos of post-conflict urban environments. However, directly leveraging such open-source imagery for machine learning applications presents several challenges. The imagery is heterogeneous in terms of scene content and neither curated nor annotated for technical use cases such as semantic segmentation. Furthermore, issues of data privacy and ethical use must be rigorously addressed when dealing with potentially sensitive imagery featuring vulnerable civilians or recognisable features.

III. *WarNav*: a benchmark for frugal segmentation of navigable zones in war scenes

A. DATTALION: a dataset of real war scene images

The DATTALION dataset [3] is a large open-source multimedia repository documenting the Russian invasion of Ukraine, launched in 2022. It consists of over 4,000 verified videos and 20,000 images, along with metadata including location, date, source, and type of event (e.g., attacks on civilian infrastructure, troop movements). The dataset is maintained by a volunteer-driven Ukrainian initiative and is primarily intended to support research, journalism, and accountability efforts related to war crimes and conflict analysis. The dataset is organized chronologically with monthly chunks. For autonomous vehicle research, only a subset of DATTALION is relevant. Many images, such as indoor scenes, nighttime photographs, or close-ups, do not provide useful information for training perception systems designed for drivable area segmentation in outdoor daytime environments.

B. Image Selection

We have first performed an initial assessment of the suitability of the DATTALION content for autonomous navigation zone detection. We have found multiple examples of outdoor road areas with partially damaged buildings or vehicles. We also found interesting scenarios such as crop field wildfires or road blast craters, which would be particularly difficult to recreate if we had to design a testing area for new image acquisition.

We then performed a progressive filtering and selection process. This filtering approach is based on past experience in artistic image competitions [1] where image quality assessment is typically performed in a few seconds during the first selection rounds. This experience has shown that selecting a few thousand images from a pre-existing repository is feasible in a reasonable time by a small dedicated team. The use of automated image preselection, such as Vision Language Models, was not considered so far, as their robustness in destructured environment was unknown.

The following methodological steps were undertaken:

- Submission of a data processing declaration in accordance with the General Data Protection Regulation (GDPR), specifying the use of encrypted hard drives and the deletion of image data upon completion of the selection process.
- Downloading of the DATTALION dataset, retaining only image files for analysis. All video files and Word documents were excluded from further consideration.
- Development of a standardized image selection protocol, including representative examples of images to be retained or discarded, based on relevance to research objectives and image quality.
- Initial filtering of the dataset through exclusion of images based on the following criteria: nighttime scenes, close-up object views, indoor settings and building facades without visible road infrastructure as only outdoor daytime scenes are relevant for our use case. Images containing blood, cadavers, or partial blurring were also removed for ethical and bias considerations.. This filtering process was conducted in parallel by team members, each responsible for a designated subset of monthly data.
- Manual review of the pre-filtered images to remove remaining outliers. This step was significantly faster than the initial filtering, thanks to the reduced volume of images requiring inspection.
- Partitioning of the monthly image subsets into training (5354 images from 8 months), validation (100 images from one month), and testing (100 images from 2 months) datasets. Note that there is no overlap between the months represented in the three sets to avoid domain leakage.

It is worth noting that several original images from the DATTALION dataset are partially blurred. These blurred regions typically correspond to cadavers or individuals whose identities were likely intentionally obscured for privacy or ethical reasons. To avoid introducing a potential bias during training, where a semantic segmentation model might learn to associate blurring artifacts with the presence of persons, we opted to discard such images. Conversely, images containing unblurred yet unidentifiable individuals were retained without modification, under the assumption that they resemble data that could be passively captured by onboard cameras of autonomous vehicles.

[1]https://www.salondaguerre.paris/

C. Semantic Classes

Based on the intended use case and the availability of this rich dataset, the set of semantic classes to be annotated was progressively refined. The following definitions were ultimately adopted:

- **Overlay**: Regions containing graphical overlays or annotations that were added post-capture. These pixels are excluded from both training and performance evaluation, as they do not correspond to real-world scene content.
- **Road**: Surfaces intended for civilian vehicular traffic, typically paved with asphalt or similar materials.
- **Drivable**: Areas that are not formal roads but are deemed traversable by military 4x4 vehicles (e.g., dirt paths, open fields).
- **Pedestrian**: Humans. Accurate detection of this class is essential for tasks related to safe autonomous navigation.
- **Vehicle**: Civilian vehicles that are potentially operable. Obstacle avoidance algorithms would consider them as potentially non-static obstacles. Damaged or abandoned car wrecks are excluded from this category.
- **Background**: All remaining regions are classified as background, encompassing areas where navigation is not feasible (e.g., buildings, vegetation, sky, rubble, blast craters or other static obstacles).

D. Annotation

Even if unsupervised techniques are foreseen to address annotation constraints, pixel annotation is necessary for performance evaluation. This annotation is performed only on validation (val) and test sets. The training dataset remains completely unannotated to emphasize the need for unsupervised learning strategies suited to real-world constraints. In practice, less than 4% (i.e., 200 among 5554) of selected images were annotated.

The annotation process began with an initial calibration phase during which a small sample of images was annotated and then discussed to clarify expectations and resolve ambiguities. The following annotation guidelines were established and agreed upon:

- **Annotation method:** Semantic segmentation was performed by manually outlining regions of interest using polygons. Each segmented pixel is assigned to exactly one semantic class; no overlapping segments.
- **Obstacle annotation:** Small debris or wreckage that could realistically be traversed by a military vehicle were not annotated individually. Conversely, blast craters are generally considered non-drivable and should be explicitly labelled as `background`.
- **Surface transitions:** Border zones between different drivable surfaces—such as the interface between asphalt and cobblestone or between paved and unpaved areas—are to be labelled as drivable if they are visually and functionally navigable.
- **Occluded road surfaces:** When dense vegetation completely obscures the underlying ground, the surface condition cannot be reliably assessed. In such cases, the

region must be labelled as `background`, as no inference should be made without clear visual evidence.

- **Sparse foreground elements:** Objects such as tree branches, leaves, or overhead cables, which do not obstruct vehicle motion but may appear in the foreground, are not annotated.
- **Vehicle versus static obstacle distinction:** The boundary between a functional vehicle and an immobile obstacle can be ambiguous, especially in war-zone imagery. The chosen criterion is based on potential operability: only vehicles that appear to be intact and potentially capable of movement are labelled as `vehicle`. Severely damaged vehicles (e.g., burned-out shells, or dismembered car halves) are treated as part of the `background`.

All test and validation images were manually annotated following this protocol. The resulting annotation masks were saved using the Cityscapes file format [8].

To assess the consistency and reliability of human annotation, a subset of 10 images from the test set was independently annotated by two additional annotators, resulting in three distinct annotations per image. The inter-annotator agreement was evaluated on all pixels: 92.3% of them were assigned identical labels by all three annotators, indicating a high level of consistency. However, 7.7% pixels showed at least one disagreement and only 0.17% pixels were assigned three completely different labels, reflecting localised interpretation ambiguities. The mean pixel-wise entropy in the dataset was relatively low (0.0492), further supporting strong annotation consistency. Pairwise Dice similarity coefficients were calculated between annotators for each semantic class. High agreement was observed in classes such as `background`, `vehicles`, `overlay` and `pedestrian` with Dice scores exceeding 0.95 across all annotator pairs. Moderate discrepancies appeared in `drivable` and `road` classes, which yielded lower Dice scores. In fact, these classes may be more prone to subjective interpretation or boundary ambiguity due to their close definitions (i.e., zones drivable by a civilian car vs. a 4x4 military vehicle). Nonetheless, these inconsistencies are not critical for the intended military application, as all affected areas still fall within the broader category of navigable space which is our primary concern. The inter-annotator agreement from this sample will serve as a benchmark for evaluating the performance of automated semantic segmentation models. Otherwise stated, we will consider the annotations having the smaller discrepancy with the two others (i.e., *Annotator 2*).

Figure 1 illustrates the distribution of pixel classes showing a strong dominance of the background class, followed by drivable areas and roads, which together account for the majority of labelled pixels. In contrast, pedestrian and vehicle classes appear significantly less frequently, which is predictable due to the war context and to their smaller size. Figure 2 illustrates the region count histogram providing insight into the spatial distribution and fragmentation of each class. While background regions remain dominant, classes like pedestrian and vehicles exhibit a higher number of small, disconnected regions relative to their pixel count. The similarity in distributions between the test and validation sets in both histograms indicates good consistency in annotation quality and dataset structure, which is crucial for reliable performance assessment.

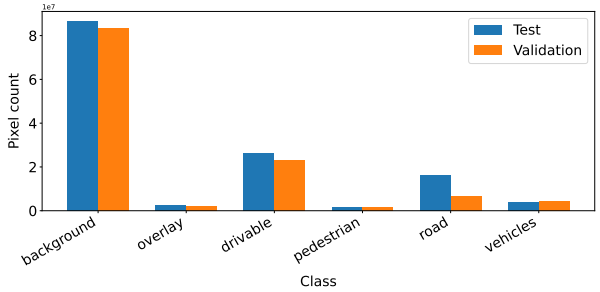


Fig. 1. Histogram of number ($\times 10^7$) of pixels per class for the test and the validation sets of *WarNav*. When ignoring 'overlay', the 5 remaining classes constitute the so-called L_5 setting used in this paper.

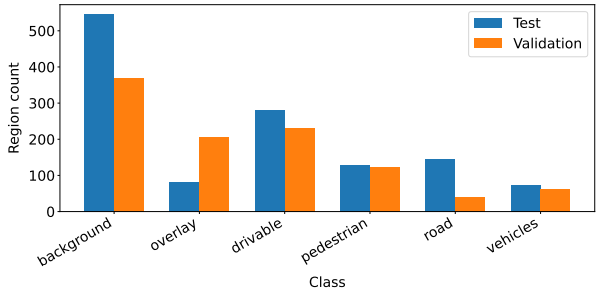


Fig. 2. Histogram of connected regions per class for the test and the validation sets of *WarNav*.

E. Dataset Open-sourcing

Selected images and annotations are available on https://github.com/CEA-LIST/WarNav. It provides DATTALION image names for the different splits and annotation masks for test and validation datasets. The original images are not shared due to licensing restrictions.

IV. Frugal baselines for *WarNav* benchmark

A. wmIoU: A new weighted mIoU suitable for WarNav

Although the mean Intersection over Union (mIoU) is the standard metric for evaluating semantic segmentation performance, it may obscure critical aspects relevant to our specific use case as it equally considers all pixels. First, since our primary goal is to ensure reliable navigability, we place greater importance on accurately segmenting regions closer to the vehicle than on distant areas. This distinction is particularly significant for the `background` class, as it encompasses both navigational obstacles such as rubble and debris, and other non-navigable regions such as sky and buildings. In our context, identifying obstacles within navigable zones is more crucial than segmenting other `background` elements, as they have a more immediate impact on navigation decisions. Secondly, we argue that accurately segmenting the inner parts of each zone is more critical than precisely delineating contours, particularly at the boundaries between `road` and `drivable` areas. To reflect these priorities, we propose a new weighted mIoU (wmIoU) that accounts for both factors,

Architecture	Backbone	#P(M)	mIoU (in %)		wmIoU (in %)	
			Cityscapes(val,L_{19})	Cityscapes(val,L_5)	*WarNav*(test,L_5)	*WarNav*(val,L_5)
DeepLabv3+ [9]	ResNet101 [10]	66	76.2	91.2	53.3	46.7
Mask2Former [11]	SwinB [12]	104	**83.3**	**93.5**	51.4	49.8
SegFormer [13]	MiT-B5 [13]	85	82.4	92.7	**61.5**	**58.1**

TABLE I

PERFORMANCES OF DIFFERENT APPROACHES BASED ON DIFFERENT BACKBONES ALL TRAINED ON THE CITYSCAPES TRAIN-SET. FOR EACH METHOD, WE PROVIDE THE NUMBER OF PARAMETERS IN MILLIONS (#P(M)), MIOU RESULTS ON CITYSCAPES VAL-SET CONSIDERING THE L_{19} AND L_5 LABELS SETTINGS, AND THE WMIOU RESULTS ON THE *WarNav* TEST AND VAL SETS. BEST RESULTS PER COLUMN ARE IN **BOLD**.

by weighting the ground truth class label map C_{gt} with a weight map W using a Hadamard product [14] (here denoted $\circ$) such as proposed by [15]:

$$wIoU = \frac{|(C \cap C_{gt}) \circ W|}{|(C \cup C_{gt}) \circ W|} \quad (1)$$

where C denotes the predicted class label map. Note that the final $wmIoU$ score is obtained by averaging the $wIoU$ values across all classes.

We draw inspiration from this work and adapt it to align with our objectives. Specifically, we introduce a distance map $D = D_1 \circ D_2$ which incorporates our two criteria:

- We consider the highest non-background pixel p_{fg} as the horizontal limit between the most critical regions that contain navigable zones (below) and the less relevant non-navigable areas (above). To reflect this distinction, we construct D_1 as two piecewise decreasing linear functions $f(a, b)$ defined by their extrema a, b, assigning greater weights to closer pixels and especially the more critical foreground ones (i.e., satisfying p below p_{fg}):

$$D_1(p) = \begin{cases} f((0,1),(p_{fg},0.8)) & \text{if } p \text{ below } p_{fg} \\ f((p_{fg},0.2),(p_{max},0.1)) & \text{otherwise} \end{cases} \quad (2)$$

- We compute a boundary distance (a.k.a. distance transform) map D_2, where for each pixel p, $D_2(p)$ is the minimum distance to a pixel of a different class normalized by the maximum value found in its connected component.

The resulting map D is then used to create a weight map $W(p) = e^{\alpha D(p)}$, to compute a $wIoU$ per class such as presented in Eq. 1, where $\alpha = 0.3$ controls the slope decay. This formulation accentuates regions farther from class boundaries, prioritizes forefront areas, closer to the camera, and especially emphasizes foreground pixels. Thus, the influence of distant background regions, which often dominate the image but are less relevant for immediate navigation, is reduced.

B. Datasets

In addition to *WarNav*, we consider three public datasets:

Cityscapes [4] is a commonly used dataset for semantic segmentation for autonomous driving. It contains 2975 finely annotated training images, and 500 validation images (val), all segmented into 19 semantic classes: L_{19}. Notably, the dataset mainly features scenes from well-structured urban environments, representing organised and structured cities.

RUGD [16] is a video dataset captured in rural and less structured outdoor environments, offering more representative samples for complex rural scenes. The original dataset is divided into 4759 train, 733 validation and 1964 test images. We modify this split to 4375 for training, 1240 for validation (val), and 1841 for testing, to (i) reduce the size of the training set for better comparability with the Cityscapes setup, (ii) ensure the inclusion of the class `water` in the training set, and (iii) minimize domain leakage across splits. The images are annotated into 24 possible class labels.

Earthquake-site database [17] (referred to as '*Earthquake*' in this paper) is a set of images depicting earthquake-related damage. It was finely segmented into 10 semantic classes such as every small crack, wreckage, or obstacle is highlighted, in contrast to *WarNav* where only bigger obstacles or blast craters obstructing military vehicle motion are considered. This dataset includes scenes of both urban and rural environments, with 686 train and 50 test images.

C. Experiments and results

In this section, we provide several baseline performances on the test and val sets of *WarNav*, analysing the influence of model architecture, backbone size, and training dataset. It should be noted that none of the models used were trained using images from *WarNav*. Instead, we report inference results from models trained on public annotated datasets. Indeed, there is an important domain gap between these datasets and *WarNav*. The presented results serve as initial baselines and provide insights into how various model characteristics influence performance in our specific application setting. Publicly available chekpoints were used to produce the results in Tables I and II. For Table III, we employed the official SegFormer code [13], with minor modifications to the dataloaders to accommodate the different datasets.

Effect of model architecture: First, we provide in Table I a comparison between various state-of-the-art segmentation models all trained on the Cityscapes [4] training set to segment images into 19 possible semantic classes (L_{19}). We chose a CNN-based model (i.e., DeepLabv3+ [9]), and two visual transformer-based (ViT [18]) approaches usually providing better results: Mask2Former [11] and SegFormer [13]. These models have different architectures, are based on different backbones (i.e., ResNet101 [10], SwinB [12] and MiT-B5 [13]), and have different memory footprints (see number of parameters #P(M) in Table I).

For each approach, we report *Cityscapes(val,L_{19})*: the *mIoU* performance on the Cityscapes *val* set segmented into L_{19}. These results illustrate the in-domain semantic segmentation performance as both training and evaluation are conducted on subsets of the same dataset with consistent class labels.

Backbone	#P(M)	mIoU (in %)		wmIoU (in %)	
		Cityscapes(val,L_{19})	Cityscapes(val,L_5)	*WarNav*(test,L_5)	*WarNav*(val,L_5)
MiT-B0	3.7	76.3	90.8	56.0	52.3
MiT-B1	13.7	78.5	91.8	54.9	49.8
MiT-B2	27.5	81.0	92.4	55.6	53.2
MiT-B3	47.3	81.7	**92.7**	58.9	55.2
MiT-B4	64.1	**82.7**	**92.7**	60.6	56.4
MiT-B5	84.7	82.4	**92.7**	**61.5**	**58.1**

TABLE II

PERFORMANCES OF SEGFORMER [13] BASED ON DIFFERENT BACKBONES ALL TRAINED ON THE CITYSCAPES TRAIN-SET. FOR EACH MODEL WE PROVIDE THE NUMBER OF PARAMETERS IN MILLION (#P(M)), *mIoU* RESULTS ON CITYSCAPES *val* SET CONSIDERING BOTH L_{19} AND L_5 LABEL SETTINGS, AND THE *wmIoU* RESULTS ON THE *WarNav test* AND *val* SETS. BEST RESULTS PER COLUMN ARE IN **BOLD**.

As anticipated, ViT-based methods significantly outperform DeepLabv3+, with larger model variants achieving higher *mIoU* scores.

Moreover, for a better comparability with the *WarNav* benchmark, we propose to map each class from L_{19} to one of the 5 classes L_5 of *WarNav* as follows ($L_{19} \rightarrow L_5$):

- road $\rightarrow$ road;
- sidewalk and terrain $\rightarrow$ drivable;
- person and rider $\rightarrow$ pedestrian;
- car, motorcycle, bicycle, truck, bus and train $\rightarrow$ vehicle;
- sky, vegetation, building, fence, wall, pole, traffic sign and traffic light $\rightarrow$ background.

As explained in Sec. III-C, we omit the `overlay` class during evaluation. We apply this mapping to all Cityscapes val prediction and ground truth segmentation maps and perform a new mIoU over the resulting L_5: Cityscapes(val,L_5). These values are higher than Cityscapes(val,L_{19}) due to the merging effect of fine-grained object classes into broader categories, which simplifies the task. For example, confusion between poles, traffic signs, and traffic lights becomes irrelevant when these are grouped into a single class. Moreover, under this mapping, the performance gap between the three evaluated approaches narrows significantly, with only a 2.3 p.p. (percentage point) mIoU difference compared to a 7.1 p.p. gap with the original L_{19} evaluation as even smaller CNN-based models succeed in performing well on this easier task.

The same mapping $L_{19} \rightarrow L_5$ is applied to predictions on test and val sets of *WarNav*, which are compared to the ground truth annotations to compute *WarNav*(test,L_5) and *WarNav*(val,L_5) respectively, using the *wmIoU* metric. In fact, as outlined in Sec. IV-A this metric is more convenient for *WarNav* dataset, contrary to other contexts such as autonomous driving in urban environments. Interestingly, the lightweight ViT-based segmentation model, SegFormer [13], achieves the best results on both sets. This could be explained by the fact that Mask2Former [11] is a panoptic segmentation model distinguishing not only the semantic concepts but also individual instances, tending to overfit to specific training instances which reduces generalization in new domains where visual patterns differ. Thus, we will use SegFormer [13] in the subsequent analyses. Note that the gap between the displayed test values and those obtained using different annotations for 10 images (see Sec. III-D for details) is always less than 0.3 p.p. *wmIoU*, which confirms the consistency of the annotations.

Effect of backbone size: Table II presents a comparative analysis of various SegFormer [13] backbones, from MiT-B0 to MiT-B5, in terms of model complexity and segmentation performance with the same evaluation settings. More details about the computational costs of each model can be found in [13]. Similarly to Table I, all models are trained on the Cityscapes *train* set to segment images into L_{19}. As expected, increasing the memory footprint leads to improved results, particularly for *Cityscapes(val,*L_{19}*)*, where mIoU rises from 76.3% for MiT-B0 to 82.7% for MiT-B4, with MiT-B5 closely following at 82.4%. When evaluating the coarser 5-class L_5 setting of Cityscapes, performance differences become less pronounced, with all models achieving scores in a narrow range between 90.8% and 92.7%. This confirms our suggestion that collapsing fine-grained categories into broader classes for Cityscapes simplifies the segmentation task, reducing the performance gap between smaller and larger models.

However, *WarNav* reveals a larger *wmIoU* gap between small and large models driven by the benchmark's complexity and the domain gap between the structured cities of Cityscapes and the severely damaged environment of *WarNav*. Indeed, *wmIoU* scores gradually improve with model size from 56.0% (MiT-B0) to 61.5% (MiT-B5) for the *test* set, and from 52.3% to 58.1% for the *val* set. Note that results are consistent across *test* and *val* sets for all models, reflecting the reliability of annotations and the representativeness of the selected images for the conflict-affected use case.

Effect of training dataset: Since Cityscapes primarily features well-structured urban environments, models trained exclusively on Cityscapes often fail to accurately segment destruction-related elements in *WarNav* benchmark (see column 3 in Fig. 3). In this section, we investigate the impact of training data by using different datasets, representing distinct types of outdoor scenes, ranging from structured urban settings to rural and destructed environments.

To ensure a fair comparison between models trained on different datasets, and since each dataset provides its unique class labels and definitions, we introduce a unified label set, L_{12}, consisting of 12 high-level semantic categories (super-classes). The labels of each dataset are mapped to this common taxonomy, as detailed in Table IV. Specifically, we retain the categories `road`, `drivable`, and `pedestrian` from L_5 *WarNav*, but refine the remaining classes as we believe that combining very distinct semantic concepts during training can harm performances. Thus, the `vehicle` category is split into three classes: `car` (civilian cars), `two wheels` (bicycles

Training Data	mIoU (in %)			wIoU (in %)	
	Cityscapes(val,L_{12})	RUGD(test,L_{12})	Earthquake(test,L_{12})	*WarNav*(test,L_5)	*WarNav*(val,L_5)
Cityscapes [4]	**89.1**	41.1	52.1	<u>58.8</u>	<u>59.9</u>
RUGD [16]	51.3	**71.5**	41.9	45.6	44.6
Earthquake [17]	61.2	39.2	<u>73.9</u>	56.0	57.9
Cityscapes+RUGD+Earthquake	<u>87.4</u>	<u>68.7</u>	**75.9**	**64.9**	**63.9**

TABLE III

PERFORMANCE OF SEGFORMER(MIT-B5) [13] TRAINED ON DIFFERENT DATASETS. FOR EACH MODEL, *mIoU* RESULTS ON CITYSCAPES(*val*), RUGD(*test*) AND EARTHQUAKE(*test*) CONSIDER THE L_{12} LABEL SETTING WHEREAS *wIoU* RESULTS ON THE *WarNav test* AND *val* SETS CONSIDER THE L_5 SETTING. BEST RESULTS PER COLUMN ARE IN **BOLD**, SECOND BEST ARE <u>UNDERLINED</u>.

and motorcycles), and `other vehicle` (larger vehicles). The broad `background` class is further divided into: `sky`, `vegetation`, `buildings`, `road obstacles` (obstacles located on the roadway), `side obstacles` (objects found outside the road area), and `water`.

Note that some inconsistencies were noticed in the annotations of Earthquake. First, grass is inconsistently annotated as either `vegetation` or `other`. As a solution, we relabel these areas as `terrain` when they are predicted as such by the SegFormer(MiT-B5) model trained on Cityscapes L_{19}. Second, in the original annotation of Earthquake, all types of vehicles are grouped under a single label. We refine this by using the same model to pseudo-label individual vehicles into: `car`, `motorcycle`, `bicycle`, `truck`, `bus`, and `train`.

L_{12} **super-class**	**Cityscapes**	**RUGD**	**Earthquake**	***WarNav***
Road	road	asphalt gravel concrete	road	road
Drivable	sidewalk terrain	dirt sand grass mulch rockbed	terrain	drivable
Person	person rider	person	person	pedestrian
Car	car	vehicle	car	vehicle
Two wheels	motocycle bicycle	bicycle	motocycle bicycle	vehicle
Other vehicle	truck bus train	-	truck bus train	vehicle
Sky	sky	sky	sky	background
Vegetation	vegetation	tree bush	vegetation	background
Buildings	building	building bridge	building	background
Road obstacles	-	log rock	cracks	background
Side obstacles	fence wall pole traffic sign traffic light	fence container table pole sign	other	background
Water	-	water	water	background

TABLE IV

MAPPING OF DATASET CLASS LABELS TO A COMMON L_{12} DEFINITION.

To assess the impact of the training data environments, we train three SegFormer(MiT-B5) models independently on Cityscapes [4], RUGD [16] and Earthquake [17], considering the L_{12} setting. The *mIoU* results on the corresponding *test*/*val* sets are reported in Table III. As expected, each model achieves the highest *mIoU* on its respective in-domain set, but exhibits significantly reduced performances on out-of-domain datasets. These large performance drops, up to 37.8 p.p. on *Cityscapes(val,L_{12})*, 32.3 p.p. on *RUGD(test,L_{12})*, and 32.0 p.p. on *Earthquake(test,L_{12})*, highlight the substantial domain gaps between these datasets and the resulting limitations in cross-domain generalization.

We further evaluate the three models on the test and val splits of *WarNav* after performing the $L_{12} \rightarrow L_5$ mapping on the predictions such as detailed in Table IV. Figure 3 illustrates qualitative results on images from the *WarNav* test set. The model trained on Cityscapes performs well in structured urban scenes (e.g., images 1 and 2), successfully segmenting classes omnipresent in such images such as vehicles and pedestrians. However, its performance degrades considerably in rural or damaged environments, where it struggles to differentiate between drivable and non-drivable areas and fails to identify road obstacles and blast craters (e.g., images 3–5). In contrast, the model trained on RUGD demonstrates better identification capacities of road and drivable areas especially when confronted with less structured scenes compared to those from urban autonomous driving settings. Yet, it is less effective in detecting finer elements such as vehicles, pedestrians, and small obstacles. Meanwhile, the Earthquake-trained model yields the best segmentation results in destructed or post-disaster environments, particularly at detecting road obstacles, even the finer ones. However, it underperforms in recognizing vehicles and people due to their limited representation in the training data.

To leverage the strengths of each individual model, we train a SegFormer(MiT-B5) model on a combined dataset comprising Cityscapes, RUGD, and Earthquake, while maintaining the unified labelling strategy. This simple yet effective approach yields a model with strong and balanced perception capabilities across diverse outdoor environments: urban/rural, structured/destructed. Notably, it performs competitively on Cityscapes and RUGD compared to single-data models and achieves the best results on Earthquake, even surpassing the model trained solely on Earthquake data. Furthermore, it strongly outperforms all previous models on *WarNav*, as shown both quantitatively in Table III and qualitatively in Fig. 3. Thus, this model took advantage from Cityscapes for pedestrian and vehicle detection, has better separation abilities between road and drivable areas thanks to RUGD, and detects road obstacles, holes and debris learned thanks to Earthquake.

V. CONCLUSION

In this work, we introduce *WarNav*, a new semantic segmentation benchmark under data annotation frugality, along

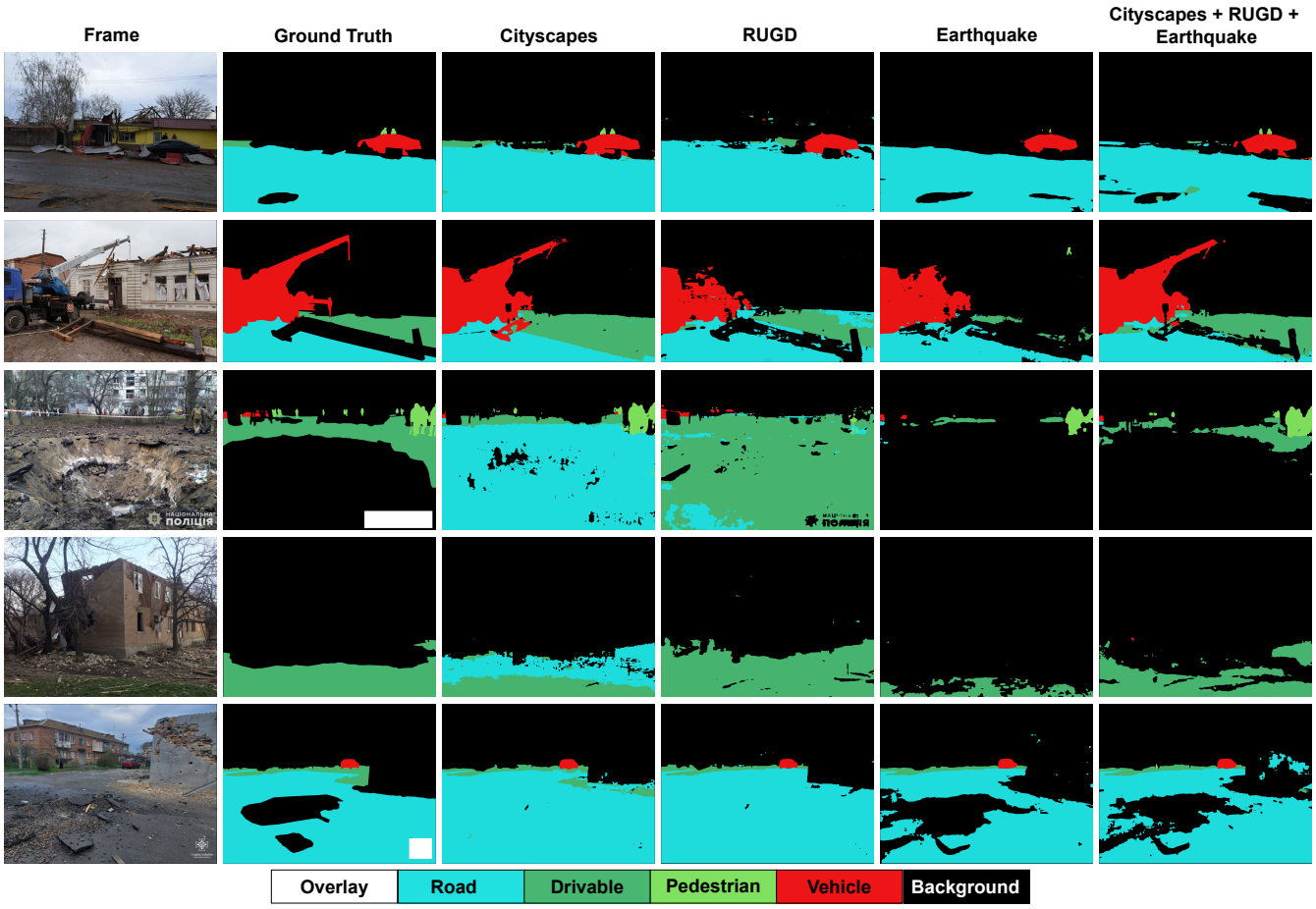


Fig. 3. Illustration of the influence of the training datasets. Columns from left to right are: test images of *WarNav*, their corresponding annotations, predictions of SegFormer(MIT-B5) trained on Citysapes, RUGD, Earthquake and the combination of the three datasets.

with baseline evaluations to assess navigability in conflict-affected areas. Our approach begins with the construction of a dataset by filtering imagery from a publicly available DATTALION repository [3]. Then, we propose a refinement of the traditional mIoU metric to better reflect the requirements of autonomous vehicle navigation in unstructured environments. Subsequently, we benchmark several baselines on *WarNav* by varying architectures, backbones and training datasets without using any in-situ images during training. Building on these results, we propose a simple yet effective solution towards autonomous navigability in hazardous zones, leveraging the diversity of available annotated outdoor environments. Our experiments focus on direct transfer of models trained on other outdoor domains to compare baseline performances. A promising direction is to employ *WarNav* training dataset as a target domain and apply Unsupervised Domain Adaptation (UDA) techniques for semantic segmentation [19], [20], thereby improving model adaptation while remaining frugal in annotations. While our study provides initial insights and solutions to enhance unmanned vehicle safety in unstructured terrains, we believe UDA-driven approaches could further improve performance. Ultimately, we hope this work will foster research in such specific environments by providing open datasets and developing frugal and robust AI models.

VI. Broader Impact

WarNav represents a semantic segmentation dataset of war-affected environments, offering a first benchmark towards developing autonomous driving systems in such challenging domains. However, the methodologies used to construct this data introduce several important considerations that merit further investigation. First, the scraping of public multimedia repositories introduces potential vulnerabilities, such as the risk of malicious remote server image manipulation. Nonetheless, this approach significantly improves researcher safety by eliminating the need for data acquisition campaigns in active conflict zones. It also improves dataset representativeness when compared to artificially constructed environments, which may inadequately capture the complexity of real-world situations. Second, the use of images sourced from public areas raises compliance challenges with the GDPR when they contain identifiable individuals, including vulnerable populations. While autonomous vehicles are expected to process similar visual data in real time to avoid pedestrian collisions, the preparation, storage, and processing of corresponding training

datasets requires explicit declaration and handling procedures under data protection regulations.

ACKNOWLEDGMENTS

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. This work was supported by the European Union under the EDF Project FaRADAI (grant number 101103386).

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-De-France Regional Council.

REFERENCES

[1] P. H. L. Rettore, P. Zißner, M. Alkhowaiter, C. Zou, and P. Sevenich, "Military Data Space: Challenges, Opportunities, and Use Cases," *IEEE COMMUNICATIONS MAGAZINE*, 2023.

[2] M. Alkhowaiter, "Detecting manipulated and adversarial images: a comprehensive study of real-world applications," Ph.D. dissertation, University of Tulsa, 2023.

[3] "Dattalion," 2022. [Online]. Available: https://dattalion.com/

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. Hassan, M. Siva, M. Lars, G. Andreas, and R. Carsten, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision (IJCV)*, 2018.

[6] C. Guettier and F. Lucas, "A constraint-based approach for planning unmanned aerial vehicle activities," *The Knowledge Engineering Review*, vol. 31, no. 5, p. 486–497, 2016.

[7] C. Guettier, W. Lamal, I. Mayk, and J. Yelloz, "Design and experiment of a collaborative planning service for netcentric international brigade command," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 2, pp. 3967–3974, Jan. 2015. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/19055

[8] "Cityscapes format annotation," 2022. [Online]. Available: https://docs.cvat.ai/docs/manual/advanced/formats/format-cityscapes/

[9] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," vol. 11211, pp. 833–851, 2018. [Online]. Available: https://doi.org/10.1007/978-3-030-01234-2_49

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," pp. 770–778, Jun. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7780459

[11] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1290–1299.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[13] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=OG18MI5TRL

[14] R. A. Horn, "The hadamard product," 1990.

[15] Y.-J. Cho, "Weighted intersection over union (wiou) for evaluating image segmentation," *Pattern Recognition Letters*, vol. 185, pp. 101–107, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865524002149

[16] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5000–5007.

[17] R. Zelek and H. Jeon, "Characterization of semantic segmentation models on mobile platforms for self-navigation in disaster-struck zones," *IEEE Access*, vol. 10, pp. 73 388–73 402, 2022.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[19] L. Hoyer, D. Dai, H. Wang, and L. Van Gool, "MIC: Masked image consistency for context-enhanced domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[20] H. Ammar, A. Loesch, C. Vannier, and R. Audigier, "Can human attribute segmentation be more robust to operational contexts without new labels?" in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 1725–1729.

From event to action: A reactive loop demonstrator for Earth Observation based on modular AI-driven components

Benjamin Francesconi
Institut de Recherche Technologique Saint-Exupéry
Sophia Antipolis, France
benjamin.francesconi@irt-saintexupery.com

Thomas Goudemant
Institut de Recherche Technologique Saint-Exupéry
Sophia Antipolis, France
thomas.goudemant@irt-saintexupery.com

Benjamin Marchand
Institut de Recherche Technologique Saint-Exupéry
Toulouse, France
benjamin.marchand@irt-saintexupery.com

Luis Palluel
Institut de Recherche Technologique Saint-Exupéry
Toulouse, France
luis.palluel@irt-saintexupery.com

Hugo Meleiro
Institut de Recherche Technologique Saint-Exupéry
Sophia Antipolis, France
hugo.meleiro@irt-saintexupery.com

Olivier Thiery
GEO4I
Creil, France
olivier.thiery@geo4i.com

***Abstract* — Operational needs in Earth Observation (EO) are increasingly demanding more responsive and autonomous systems, particularly for security and defense applications. This requires new architectures able to shorten the decision–action cycle through real-time event detection, adaptive tasking, and intelligent onboard analytics. The IRMA project, led by IRT Saint Exupéry, develops Artificial Intelligence (AI)-based technologies for mission planning and data processing of EO constellations (both on the ground and onboard satellites) to enhance reactivity and decision-making in realistic end-to-end scenarios. This paper presents the IRMA demonstrator, a modular platform emulating a complete EO system and integrating advanced technologies such as the adaptive multi-agent scheduler ATLAS2 and a YOLOX-based ship detection pipeline. It validates autonomy, robustness to operational constraints, and clarity of outputs for human operators, three key challenges for security / defense applications. The demonstrator executes fast-paced, end-to-end scenarios on real data, offering an engaging and operationally relevant user experience. It provides a testbed to mature AI building blocks, assess system-level reactivity, and explore the architecture of future EO systems combining ground and onboard intelligence. Its design supports modularity, standardized APIs and real-time visualization, and will soon integrate embedded processing hardware to enable hybrid ground/onboard workflows in line with security and defense requirements for autonomy and frugality.**

Keywords — Earth Observation, Reactive Systems, Mission Planning, Multi-Agent Systems, AI for Space, AI for Security and Defense, Onboard Processing, Edge Computing, Demonstrator, System Autonomy, Maritime Surveillance

I. INTRODUCTION

The Earth Observation (EO) domain is undergoing a profound transformation, driven by the growing demand for more responsive, autonomous, and intelligent systems. In both civilian and defense contexts, users now expect satellite systems to move beyond data delivery and provide timely, actionable insights like detecting, interpreting, and reacting to events such as natural disasters, illegal activities, or military threats within minutes rather than hours. While current constellations already produce tens of terabytes of imagery daily [1], traditional EO workflows often introduce delays of several hours, sometimes a full day, before information reaches decision-makers.

This latency is increasingly incompatible with time-critical missions. In defense, space-based Intelligence, Surveillance, and Reconnaissance (ISR) relies on fast detection and re-tasking. The European Defense Fund's SPIDER project directly addresses this challenge by promoting autonomous planning, short revisit cycles, and minimal end-to-end latency [2], [3]. In the civil domain, NASA's Earth Science to Action strategy similarly calls for reducing the gap between observation and response, prioritizing decision-ready information [4]. These converging priorities are further amplified by the rise of New Space and the growing availability of agile, multi-sensor constellations, reinforcing the need for integrated, low-latency decision-action loops — both on the ground and onboard satellites.

The necessary transformation to meet this challenge impacts all components of EO systems. In particular, institutional, commercial, and industrial strategies increasingly converge on a set of key enabling technologies:

- *Artificial Intelligence (AI)*, for high-level reasoning and interpretation of multi-modal data (optical, radar);
- *Edge Computing on-board satellites*, enabling early detection, filtering/prioritization and autonomous decision-making (e.g. triggering follow-on actions), as demonstrated by missions like Phi-Sat 2 [5] and CogniSAT-6 [6];
- *Inter-operability and orchestration*, to federate heterogeneous multi-mission assets and coordinate them under tight timing and mission constraints;
- *Low-latency and seamless communication infrastructures*, including Ground Station as a Service and optical or radiofrequency Inter-Satellite Links (ISL), to enable real-time feedback loops and ensure global system reactivity.

These technological directions are echoed in the strategic roadmaps of major space stakeholders—including the European Union, ESA, CEOS, and NASA—as highlighted in recent reports and white papers [7][8] [9][10][11][12][13][14].

Collectively, these efforts signal a structural shift from linear, siloed EO systems toward distributed, intelligent,

and reactive architectures. Such a shift is essential to meet the evolving requirements of both civilian operations and time-critical defense applications. Fig. 1 illustrates this shift from traditional architectures to the next generation of responsive EO systems.

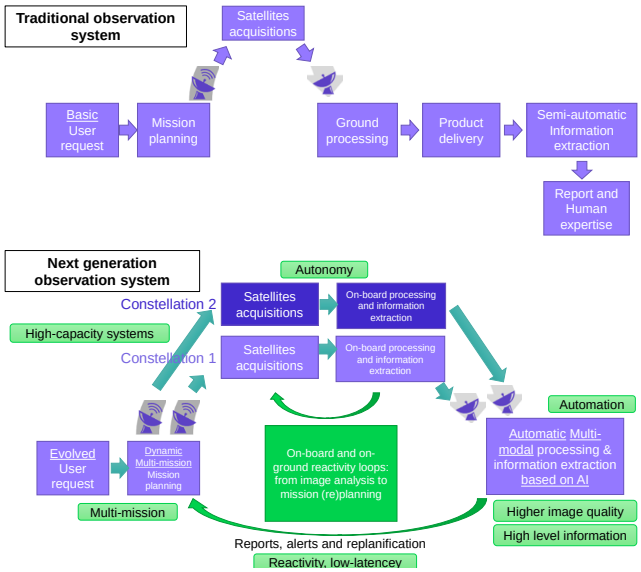


Fig. 1. From traditional EO systems to intelligent, reactive architectures.

The IRMA project (Image processing for a Responsive Mission with AI) led by IRT Saint Exupéry [15], contributes to this transition through AI and edge computing core technologies by developing a suite of AI-based technological building blocks for intelligent and reactive mission planning and data processing, on ground as well as on board.

IRMA is also developing a demonstrator in order to validate and quantify, through realistic and illustrative end-to-end scenarios, the added value of these technologies in terms of system reactivity and autonomy. This demonstrator is a modular hardware and software platform that integrates IRMA technologies into an architecture emulating the main operational components of EO systems.

The idea of more agile and intelligent EO architectures has been discussed in the scientific community for at least a decade. In 2015, Golkar presented a federated Satellite systems paradigm [16] envisioning heterogeneous spacecraft cooperating by sharing resources and services to enhance efficiency and resilience. Denis et al. [17] later examined potential disruptions in Earth Observation systems and markets, highlighting how New Space constellations, data-as-a-service models, and platform-based business approaches could fundamentally reshape EO value chains. More recent works have proposed mission and system architectures supporting persistent and multi-sensor monitoring [18], or demonstrated how autonomous onboard intelligence can improve the exploitation of high-dimensional EO data [19].

In parallel, several European initiatives are translating these concepts into concrete system developments. DOMINO-X [20] is a collaborative effort to modernize EO ground segments through modular building blocks and standardized interfaces. Building on that groundwork, DOMINO-E introduces a multi-mission federation layer to orchestrate sensors across mission boundaries and optimize reactivity through advanced scheduling [21]. Other projects also illustrate this paradigm shift. For example, LEONSEGS [22] explores federated multi-mission ground segments, CALLISTO [23] integrates Copernicus DIAS (Data and Information Access Services) data with heterogeneous sources through AI and big data processing; and RapidAI4EO [24] develops spatiotemporal AI models for high-cadence land monitoring. At the same time, on-board AI demonstrations (ESA Φ-sat-1/-2, OPS-SAT) show practical pathways to filter, prioritize, and act on data at the edge, from cloud-screening [25] to anomaly detection experiments in-orbit [26].

While these initiatives are actively addressing future Earth observation system needs in Europe, most efforts still either work on defining high-level flexible architectures or target isolated technological bricks. The IRMA demonstrator takes a complementary and original approach by bridging system architecture and operational concepts with the integration of concrete, state-of-the-art technologies enhancing key system functions both on ground and on board. It provides a unique environment to assess how these technologies interact within full-system workflows and how they jointly contribute to complex performance indicators such as system reactivity and autonomy.

The remainder of this paper is structured as follows. Section II introduces the main requirements of the IRMA demonstrator and the adopted development approach. Section III presents the demonstrator architecture and the integration of its core components. Section IV details the AI-based technological building blocks integrated into the system. Section V illustrates a representative use-case scenario, highlighting the reactivity loop and dynamic coordination between modules. Finally, Section VI concludes the paper by emphasizing the demonstrator's contributions and outlining perspectives for future developments.

II. REQUIREMENTS AND DEVELOPMENT APPROACH

A. Operational Context

The main goal of the IRMA demonstrator is to illustrate, through "live" demonstration sessions, reactive system loops on realistic scenarios, where high-level User Requests (UR) trigger adaptive acquisitions, processing and reprogramming actions based on AI-driven insights.

An effort was undertaken to identify Earth Observation use-cases requiring high responsiveness, in which traditional EO systems fall short due to long processing and reaction cycles. This analysis is summarized in TABLE I and provides a foundation for aligning system functionalities with real-world operational needs.

At maturity, the IRMA demonstrator is expected to support complex scenarios such as the following:

A high-resolution multispectral satellite is tasked to acquire images over a conflict area. Thanks to its on-board processing capabilities, it detects the spectral signature of polymer materials (e.g., plastics) within a densely vegetated area indicating a potential camouflage material. An alert and a lightweight report are immediately transmitted to the ground through a low bandwidth channel. On board, the alert also leads to the prioritization of that image's downlink on the next ground-station overpass.

On ground, the alert automatically triggers the urgent re-tasking of a high-resolution radar satellite to acquire a follow-up image over the same area. Upon reception, the radar image is processed and reveals a metallic echo at the exact location previously flagged, confirming the likely presence of a concealed material. Further exploitation of the radar signature may allow for coarse classification of the object (vehicle, structure or other material), depending on image characteristics and target dimensions.

This example highlights how the demonstrator bridges operational needs with enabling technological capabilities.

TABLE I.
EARTH OBSERVATION USE-CASES AND THEIR TYPICAL REACTIVITY NEEDS

ID	Theme	Expected Latency
1	Maritime surveillance – Oil spills	< 1h
2	Maritime surveillance – Algae, sediments	< 3h
3	Maritime surveillance – Illegal activities	< 1h
4	Port or Airport monitoring	< 1h
5	Natural disaster (Earthquakes, Floods, Hurricanes…) / War zone monitoring	< 30min
6	Wildfires	< 15min
7	Search & Rescue	< 15min
8	Monitoring of critical or military industrial sites	30min – 1 day
9	Monitoring of large areas (e.g., deforestation, borders)	30min – 1 day
10	Soil analysis / Precision farming	< 24h
11	Camouflage detection	30min – 6h
12	Air quality monitoring – Methane	< 1h

B. Key System-Level Requirements

The illustrative scenario described in the previous paragraph is representative of the end-to-end reactivity that IRMA aims to support. To achieve this, the demonstrator is designed to integrate AI technologies into a simulated EO system comprising at least the following components and interfaces:

- Space segment that is configurable with the number of satellites and their main parameters: agility, orbit type (sun-synchronous (SSO), inclined, etc.), and payload modalities (optical, IR, SAR, hyperspectral);
- Smart mission planning function;
- On-ground and on-board data processing;
- Reactivity service, to close the loop between data processing and mission planning;
- Simulation of communication links, with configurable number and location of ground stations (including Ground Stations as a Service), as well as additional links such as low-bandwidth RF channels or Inter-Satellite Links (ISL).

During scenario execution, IRMA AI technologies must be run in real time on real data. On-board processing must be executed on a real edge device with embedded hardware. For live demonstrations, the system must compress the execution of an operational scenario (normally spanning 6–24 h) into less than 10 min, with real-time visualization of key events and performance metrics.

The demonstrator shall showcase as many of the following capabilities as possible:

Event tracking and automatic reprogramming through a feedback loop between image analysis (on-ground or on-board) and mission planning.
Optimal constellation planning, maximizing mission capacity (number of images), revisit frequency, and information freshness.
Mission reactivity for dynamic planning of urgent requests.
Semantic information extraction from mono- and multi-modal images via ground-based processing.
Semantic information extraction from mono-modal images via on-board processing.
Selective processing (on-ground or on-board) depending on acquisition request characteristics.
Ability to follow the user request status from definition to completion.
Prioritization of satellite downlink schedules based on urgency and the semantic content of on-board processed images.
Ability to update on-board processing algorithms during the system's lifetime.
Automatic backup acquisition to replace failed attempts (e.g., due to weather or anomalies).
Capability to program a multi-mission system.

C. Development Challenges and Strategy

Designing such a demonstrator poses several key challenges, including the integration of heterogeneous software bricks of varying levels of maturity and origin (R&T developments, industrial partners and legacy projects), their interoperability within a streamlined yet representative EO system architecture, and the need to combine real-time execution with offline or embedded components while ensuring consistent interface management and temporal synchronization. Additional challenges include providing a positive and engaging User Experience (UX) during live demonstrations, as well as ensuring maintainability and modularity for future expansions.

To address these challenges, the team adopted an agile, incremental development approach, allowing step-by-step integration and testing of components as well as iterative refinement based on user feedback and UX evaluations. A model-based systems engineering (MBSE) methodology using Capella [27] was also employed to support high-level architectural specification, functional decomposition, and traceability of system requirements. The demonstrator architecture was aligned with the principles of DOMINO-X [20], which defines a modular ground segment for next-generation EO systems. This architecture has been tailored to the IRMA demonstrator scope, focusing on components where AI brings operational value.

III. SYSTEM ARCHITECTURE

A. Software and Functional Architecture

In its current version, the IRMA demonstrator emulates a realistic EO architecture, as illustrated in Fig. 2. It relies on a central orchestrator designed to coordinate the simulation timeline, manage time-sensitive interactions, and trigger key events (e.g., acquisitions, downlinks, processing). This mechanism ensures deterministic temporal control and

smooth integration, while remaining consistent with the principles of DOMINO-X promoting modular, event-driven and loosely coupled architecture. Our approach and used technologies also echoes NASA's NOS-T (New Observing Strategies Testbed) prototyping platform for distributed space missions [28].

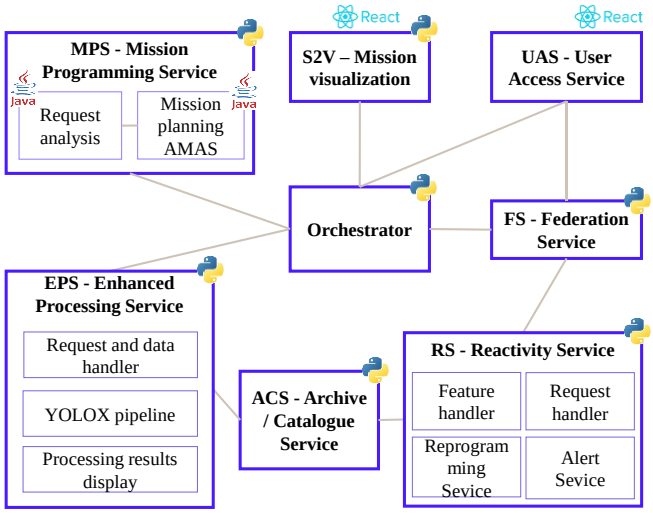


Fig. 2. Simplified overview of the high-level software architecture of the current IRMA demonstrator. The primary programming language used for each component is indicated by its corresponding icon.

The architecture includes the key components defined in DOMINO-X, complemented by a few additional modules (indicated with a *) specific to the demonstrator:

- User Access Service (UAS): The main human-machine interface for defining and visualizing user requests as well as the scenario timeline.
- Mission Programming Service (MPS): Performs meshing and analyzes the feasibility of an acquisition request, then uses an AI-based Adaptive Multi-Agent Planner (AMAS) for dynamic scheduling.
- Enhanced Processing Service (EPS): Performs AI-based image analysis in response to the user request (e.g., ship detection with a YOLOX model).
- Reactivity Service (RS): Manages event follow-up and makes decisions (such as triggering an alert or (re)programming an acquisition) based on comparison between EPS outputs and user request criteria (rule-based engine).
- FS (Federation Service): The central orchestrator in the Domino-X architecture, responsible for unified management of user requests and workflows across multiple systems. In the IRMA demonstrator, it is implemented as a simplified function focused on request handling.
- Archive & Catalog Service (ACS): Indexes raw and processed products using OGC STAC standards. Implemented with minimal functions supporting other components.
- Orchestrator (*): Drives the simulation, coordinates components, manages the mission timeline, and enables observability.
- Mission Visualization Tool (*): A Cesium-based application, referred to as S2V (Scenario to Visualization), acting as the main HMI for dynamic scenario rendering. It provides real-time visualization of satellite operations, orbital tracks, and ground stations in both 2D and 3D environments.

The demonstrator leverages standardized, well-established technologies. All components are containerized with Docker and expose interoperable REST/OGC interfaces for smooth integration and scalability. Communication between services relies on modern frameworks such as FastAPI and MQTT, enabling real-time interaction, responsiveness, and advanced visualization. The entire stack supports automated deployment through Docker Compose or Swarm, reinforcing maintainability and enabling future extensions to more complex or operational deployments. Although the current demonstrator focuses on ground-based components, the architecture is designed to integrate on-board processing modules via a dedicated compute board in the next release.

B. Hardware architecture

Physically, the demonstrator is hosted in a modular flight case with three interconnected hardware stations, each with its own display representing a key part of the simulated EO system, as shown in Fig. 3.

Allocation of system components and HMI to Hardware components is shown in TABLE II.

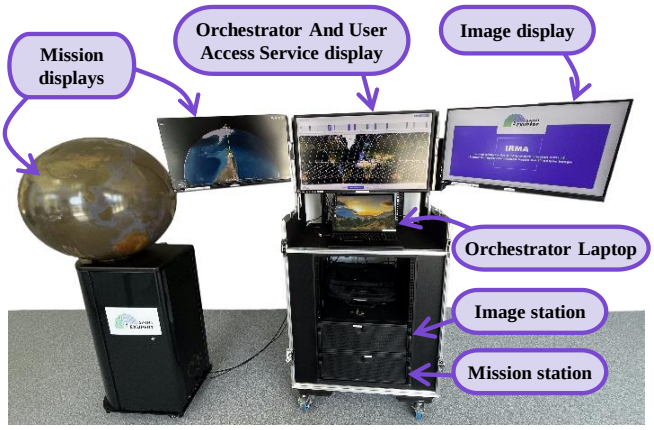


Fig. 3. Hardware setup of the IRMA demonstrator. Upcoming versions will include an embedded target to emulate on-board processing.

TABLE II.
HARDWARE LIST AND SOFTWARE COMPONENTS / HMI ALLOCATION

Hardware station	Software components	Visual Interface
Mission	MPS, S2V	AMAS internal acquisition request status, ***global vision of the constellation***.
Image	EPS, ACS, RS	***Images and outputs from EPS*** (e.g. detection bounding boxes).
Orchestrator	Orchestrator, UAS, FS	***Scenario timeline***, ***User Request selection/validation*** and follow-up, alerts, reports and suggested reprogramming request, reactivity dashboard.
Spherical screen	S2V	Global vision of the constellation.
Edge target	(Upcoming): On-board processing and reactivity.	

[a.] Interactive HMIs in ***bold italics***

The demonstrator also includes a spherical screen displaying Earth and constellation dynamic evolution (from

S2V) to increase UX. Additionally, in a close future an embedded hardware target will be connected to the demonstrator enabling real-time on-board processing for illustration of new, more reactive, operational scenarios.

IV. Technological Building Blocks

The demonstrator integrates key AI-based technologies into an architecture emulating the main operational components of EO systems. It validates their integration, illustrates their added value within a responsive system loop, and supports TRL progression through interoperable, standardized interfaces. It enables system-level evaluation of autonomy, alignment with operational constraints, and clarity of outputs for human decision-making, three central challenges for security / defense applications.

The IRMA project develops multiple AI-based technological bricks at varying maturity levels, including multi-modal image processing (e.g., object detection, segmentation, image enhancement), representation learning (e.g., image retrieval, captioning), foundation models, and unsupervised anomaly detection on both imagery and time series. For mission planning, a legacy Adaptive Multi-Agent Planner (AMAS) is being upgraded. The project also investigates several embedded platforms (AMD, Intel, Nvidia), leveraging vendor-specific toolchains to deploy and benchmark IRMA algorithms, with a focus on improving the robustness of AI models when processing raw remote sensing data directly on board satellites.

In its current version, the demonstrator integrates two flagship AI-based technologies, described in the following paragraphs: adaptive and reactive mission planning with AMAS, and ship detection and recognition with YOLOX.

A. *Adaptive and Reactive Scheduling with AMAS*

A central component of the IRMA demonstrator is the Mission Programming Service (MPS), which handles the planning and scheduling of satellite acquisitions. This component integrates an AI-based planner grounded in the Adaptive Multi-Agent System (AMAS) paradigm [29]. More specifically, the AMAS implemented in the MPS is a redesigned version, called ATLAS2, which enhances the responsiveness of Earth observation systems by supporting feedback loops from image analysis to mission planning [30].

In contrast to traditional greedy algorithms still widely used in operational systems, ATLAS2 enables real-time and dynamic planning, supporting the insertion of last-minute or high-priority requests without restarting the entire planning process. The agent-based design models satellites, user requests and acquisitions as cooperative agents capable of negotiating conflicts and adapting to evolving constraints. More precisely, the main intelligence of the multi-agent system lies in the way acquisition agents negotiate with each other to resolve the non-cooperative conflict situation, perceived by a satellite agent, where a required time slot is already booked by another acquisition agent. The negotiation is based on the criticality of the request (e.g. its priority) and on the scheduling cost of this request across all available satellite resources. This flexibility makes the system particularly well-suited for reactive Earth Observation scenarios such as disaster response, environmental monitoring, or maritime surveillance.

In [30], benchmarks on realistic scenarios with agile satellites constellations in demonstrate that ATLAS2 can lead to up to a 30% improvement in the number of planned requests compared to a state-of-the-art hierarchical greedy algorithm, particularly in complex, resource-constrained situations (e.g., two-satellite systems with thousands of requests). It also shows faster and more robust integration of urgent requests, as illustrated in Fig. 4, typically re-planning within less than one minute, and resolves scheduling conflicts more effectively through local negotiation mechanisms.

Finally, ATLAS2's decentralized nature provides inherent scalability to multi-constellation systems, and its “any-time” behavior makes it suitable for use in continuous planning loops with feedback from image analysis. These properties are key enablers for future architectures where on-ground and on-board mission planning must coexist and interact seamlessly.

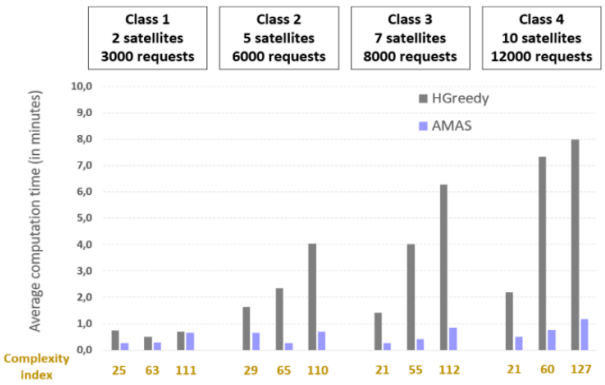


Fig. 4. Time to plan an urgent request with the AMAS algorithm (purple) compared with HGreedy (gray) for scenario classes of increasing complexity (excerpt from [30])

B. *Ship detection and recognition with YOLOX*

Another core component integrated in the IRMA demonstrator is the on-ground Enhanced Processing Service (EPS), which hosts AI-based image analysis capabilities. In the current setup, this service includes a real-time ship detection and recognition module based on YOLOX, a member of the “You Only Look Once” family of detectors [31], deployed on standard GPU-based hardware.

This Convolutional Neural Network (CNN) module builds upon prior work carried out by IRT Saint Exupéry in the CIAR project and presented at CAID in 2022 [32], where a YOLOv3-based solution had been implemented and assessed for its suitability for on-board deployment and low-latency detection of vessels from high-resolution satellite imagery. Building on this experience, a YOLOX-S network (S for “Small” backbone) was selected for the IRMA demonstrator due to its improved balance between detection accuracy, model size, and computational efficiency. This exploration of lightweight embedded models directly addresses the need for frugality and constrained-resource environments, a critical concern in security and defense systems.

YOLOX was trained and validated on a unique, high-quality dataset specifically created for IRT, consisting of over 24,000 annotated ships across 46 classes, including small vessels, military ships, and commercial cargo ships. The dataset, derived from high-resolution (30-50 cm GSD) MAXAR imagery, was labeled by expert photointerpreters from GEO4I. It contains 24,000 patches of 640 × 640 pixels.

YOLOX detection/recognition and hardware performance results are summarized in TABLE III. Evaluation shows that YOLOX-S achieves F1-scores above 40% and a mean

Average Precision (mAP) of around 30% on unseen test images. These global figures are penalized by lower performance on underrepresented ship classes, but despite this imbalance, excellent precision and recall (both above 90%) are achieved for dominant categories such as fishing vessels, sailboats, and leisure craft, with promising generalization to less represented types. Overall, this level of performance is considered sufficient for the demonstrator.

Inference tests on an AMD FPGA confirm that YOLOX-S is lighter and faster than YOLOv3, making it a suitable candidate for future integration in the demonstrator's embedded hardware.

TABLE III.
YOLOX PERFORMANCE SUMMARY ON OUR CUSTOM SHIP DATASET

Complexity				
Model size	65 MB			
Complexity	26.8 Gflops			
Performance on compute station				
Performance on test dataset	Precision: 41.5%	Recall: 41.6%	F1-Score: 41.55%	mAP: 29.7%
Performance on Xilinx ZCU104 FPGA (deployed with VITISAI 3.0)				
Performance on test dataset	Precision: 38.1%	Recall: 37.5%	F1-Score: 37.8%	mAP: 27.6%
Hardware performance (batchsize=1)	Latency: 29ms		Throughput: 14Mpx/s	

To meet the needs of the demonstrator scenario, which must operate on full real images (and not only small patches from datasets), YOLOX has been integrated into a complete ship detection pipeline capable of processing large remote sensing images. The pipeline includes image tiling and dynamic range adaptation as pre-processing steps, and detection map reconstruction at image scale as post-processing.

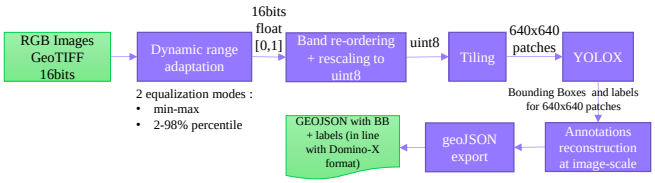


Fig. 5. Complete ship detection and recognition pipeline based on YOLOX integrated into the EPS.

Once integrated into the EPS, the delay between the start of the processing pipeline and the display of the results remains under one minute for demonstration images ranging from 100 to 700 megapixels. This latency is acceptable for demonstration purposes, with most of the time being spent on launching the YOLOX Docker container and handling data transfers.

V. ILLUSTRATIVE SCENARIO: MARITIME SURVEILLANCE

A. Use-case Selection and Simulated EO System

In this section, we illustrate the demonstrator's execution on a representative scenario. Among the use-cases listed in TABLE I, illegal fishing detection was selected as the first demonstrator scenario for several reasons:

- It requires rapid response loops for effective interdiction and acts as a proxy for time-critical security / defense missions.
- It builds on existing IRMA capabilities and previous projects, notably the YOLOX-based ship detection models and annotated datasets [32].
- It produces visual and interpretable outcomes, useful for validation and demonstration purposes.

A realistic EO system was configured alongside the use-case selection, composed of three satellites and two ground stations. The space segment includes one Very High Resolution (VHR) optical satellite (30 cm GSD) in Sun-Synchronous Orbit (SSO) and two High-Resolution (HR) optical satellites (70 cm GSD) in inclined orbits to increase revisit frequency at mid-latitudes. All satellites have high agility. The ground segment includes uplink/downlink stations in Kiruna (Sweden) and Toulouse (France). The scenario spans a 24-hour period from June 21 to June 22, 2025. This is summarized in TABLE IV.

To maintain operational realism, the orchestrator injects latencies related to telecommunications and non-simulated operations (e.g., primary ground processing for sensor correction and georeferencing). These values are predefined, based on typical performance in EO systems and operational partner feedback.

TABLE IV.
MAIN PARAMETERS OF THE SIMULATED SYSTEM

Parameters	**Value**
# of Satellites	3 satellites with high agility
Satellite 1	VHR optical @30cm GSD, 19km swath
Satellite 2 & 3	HR optical @70cm GSD, 19km swath
Orbits	- Sat. 1: 550km; Sun-Synchronous (SSO) - Sat 2&3: 550km; Inclined
Scenario duration	24h from 21/06/2025 to 22/06/2025
Ground stations	Kiruna (SWE) + Toulouse (FRA) (both for uplink and downlink)

At scenario start, the system is pre-loaded with 1,000 background acquisition requests of type SPOT (19×19 km) or STRIP (19×[20–200] km), distributed globally. Up to 2,000 additional requests may arrive during execution. These requests are not tied to specific use-cases but simulate a realistic workload and stress-test for the ATLAS2 multi-agent planning system.

In parallel, several high-priority User Requests (URs) represent the selected use-case. Their format, inspired by DOMINO-X [20] preliminary definition, has been largely improved to cope with the needs of our scenarios (in terms of reactivity and processing needs) and with our mission planning tool interfaces. When selected by the user, a UR triggers a full end-to-end reactive loop, activating the different AI technological bricks within a realistic operational context, thereby validating their proper functioning and illustrating their operational relevance.

During scenario execution, the user can freely adjust time acceleration. However, all IRMA technologies are executed in real time to showcase their actual performance, requiring strict synchronization by the orchestrator.

B. Scenario Execution and Functional Chain Validation

In this maritime surveillance scenario, the reactive loop is initiated when the user selects and validates a pre-defined UR

in the UAS. This activates the full end-to-end functional chain of the demonstrator, summarized in Fig. 6. An alert is triggered if at least one fishing vessel is detected in the image (assuming the area is a prohibited fishing zone).

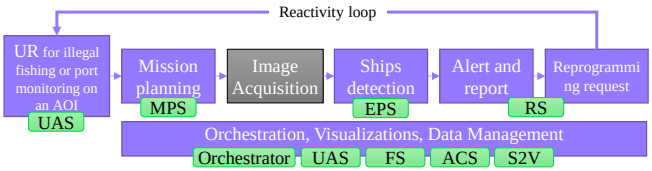


Fig. 6. Simplified functional chain of the maritime surveillance scenario

Illustrative outputs from the demonstrator, generated during an illegal fishing detection scenario over the Golfe du Morbihan (France), are shown in Fig. 7 on the next page. The screenshots illustrate, in order (left to right, bottom to top):

- *UAS: Locations of predefined User Requests.* Initial state of the scenario, with the system already processing background requests and waiting for a high-priority one.
- *UAS: Selection of an illegal fishing UR.* The user chooses among several predefined requests, each triggering a reactive loop that showcases the role of AI in enhancing responsiveness.
- *UAS: Selected UR parameters.* User-defined acquisition, processing, and reactivity parameters are displayed.
- *UAS: UR follow-up interface.* Once a UR is selected from the HMI, it is sent to the Federation Service (FS), which dispatches its elements to other components. At each major event, the UR status and scenario timeline are updated in the UAS.
- *MPS: ATLAS2 acquisition requests internal state.* It shows how the mission planner schedules acquisitions, prioritizing high-priority requests.
- *S2V: Dynamic mission visualization.* The user can follow the scenario's space segment activities in real time. When a satellite passes over the ground station, the orchestrator simulates plan upload and data download while S2V shows corresponding communications with ground stations.
- *EPS: Ship detection and recognition with YOLOX.* Once the UR image is acquired and downloaded, the EPS retrieves and processes it, displaying both the image and detection results.
- *RS/UAS: Detection report and reprogramming request.* When a ship is detected in a non-fishing area, the RS generates an alert, a report, and a suggested reprogramming request, all displayed in the UAS for user validation.

This scenario validates the end-to-end integration of IRMA technologies and demonstrates their relevance in a realistic maritime surveillance context. It also shows how AI-driven autonomy can accelerate decision-making.

VI. Conclusion and Perspectives

The IRMA demonstrator provides a unique environment to validate reactive system loops in Earth Observation, bridging system architecture, operational concepts, and state-of-the-art AI technologies. It offers a tangible and operationally relevant platform to mature technologies, validate functional integration, and test interoperability between components. These objectives align with European strategic initiatives such as the Earth Observation Governmental Service (EOGS), currently under ESA and EU study contracts, and the upcoming ERS-EO program, both aiming to enable resilient and responsive EO capabilities for security and defense applications.

By integrating concrete capabilities such as adaptive multi-agent planning and real-time ship detection with YOLOX, IRMA demonstrator shows how autonomous decision loops can be implemented and evaluated under realistic conditions. It thus accelerates the maturation of key AI components, enforces standardized interfaces, and highlights their operational value through interpretable, user-oriented outputs. Future developments will extend its scope to additional use-cases, multi-sensor configurations, and onboard intelligence.

Lessons learned from IRMA also address broader security / defense challenges. The demonstrator illustrates how autonomy can be enabled through closed-loop reactivity, how robustness can be strengthened by testing AI on representative scenarios, and how explainability can be enhanced by providing transparent outputs at every stage of the loop; all aspects fully aligned with the challenges emphasized by CAID 2025. The forthcoming integration of FPGA platforms also contributes to frugality, a critical requirement for space-based and defense-oriented applications.

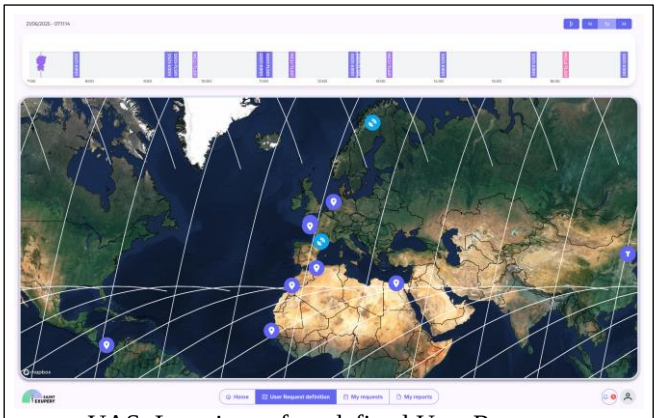
In addition to serving as a communication and integration tool, the demonstrator paves the way for a future end-to-end performance simulation framework. Such a tool is increasingly needed to quantify reactivity performance, now a key decision factor for institutional and commercial EO users. Unlike classical metrics such as revisit time, which only reflect acquisition capability, reactivity is a system-level metric that depends on the coordinated behavior of satellites, ground segments, communication links, and processing both on board and on ground. Enhancing global system reactivity therefore requires progress across almost all EO system domains and is intrinsically tied to automation and autonomy.

As the next steps unfold, the IRMA demonstrator will continue to act as a catalyst for advancing the design and evaluation of intelligent, responsive EO systems, while contributing to the development of next-generation autonomous security / defense architectures.

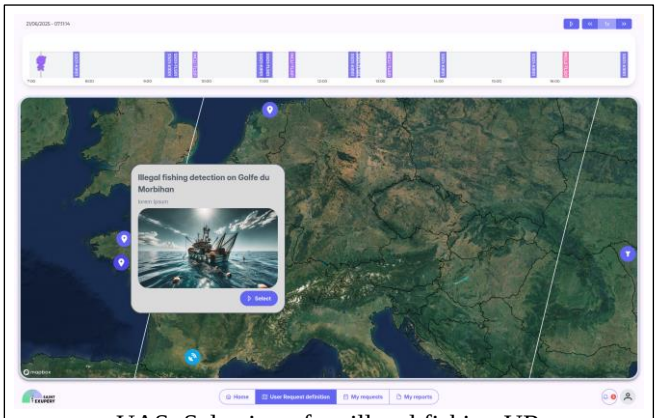
Acknowledgment

The authors would like to thank the industrial and academic partners of the IRMA project: Thales Alenia Space, Activeeon, Geo4i, JoliBrain, and University Côte d'Azur. They also extend their gratitude to the European Space Agency for enabling them to participate in the Φsat-2 in-orbit demonstration, as well as to the Centre National d'Études Spatiales for their support through postdoctoral researcher funding.

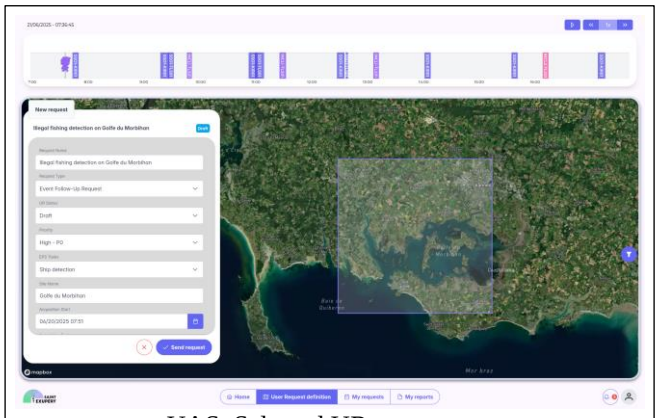
This work was partially funded through the French *France 2030 Programme*.



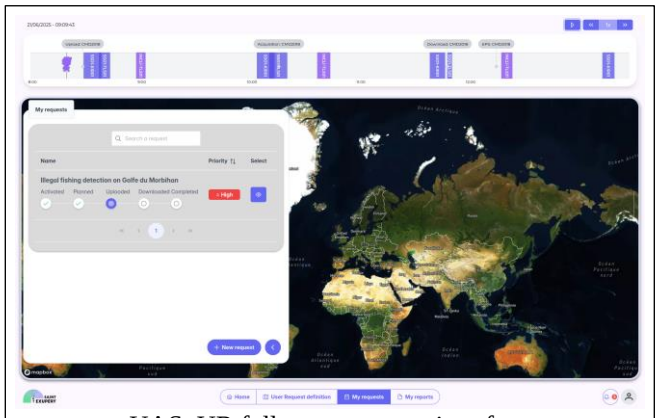
UAS: Locations of predefined User Requests



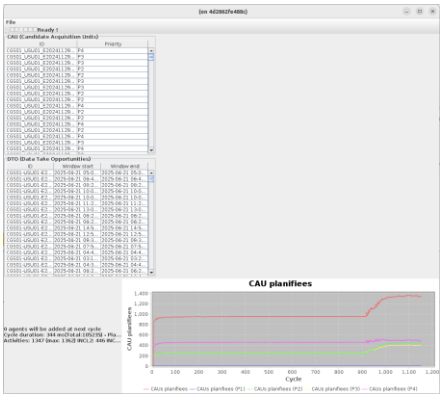
UAS: Selection of an illegal fishing UR



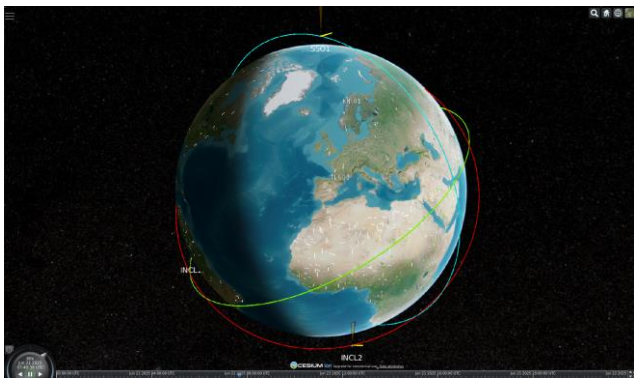
UAS: Selected UR parameters



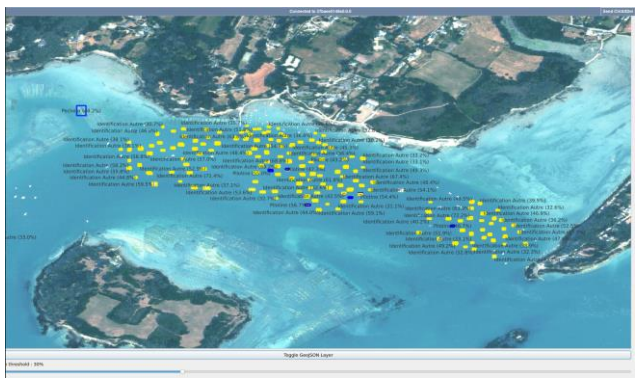
UAS: UR follow-up request interface



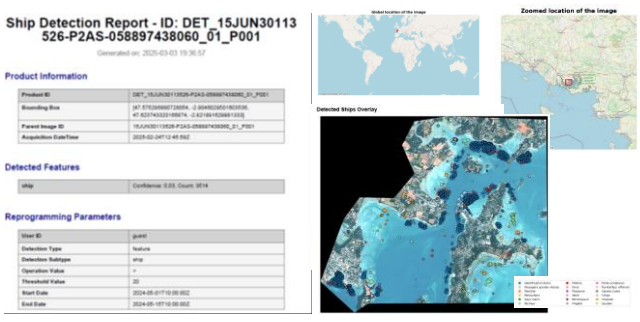
MPS: ATLAS2 acquisition requests internal state



S2V: Dynamic Mission visualization



EPS: Ship detection/recognition results with YOLOX



RS/UAS: Detection report and reprogramming request.

Fig. 7. Simplified functional chain of the maritime surveillance scenario. (Satellite images: Maxar Imagery Product © 2015 Maxar Technologies Technologies.)

REFERENCES

[1] European Space Agency, “Copernicus Sentinel Data Access Annual Report 2023,” Issue 1.0 Rev. 1, Aug. 7, 2024. [Online]. Available: https://sentinels.copernicus.eu/-/ninth-copernicus-sentinel-data-access-annual-report

[2] European Commission. SPIDER – Space-based Persistent ISR with Autonomous Retasking. Grant Agreement No. 101121113. European Defence Fund (EDF), 2022. [Online]. Available: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/projects-details/44181033/101121113/EDF

[3] European Commission, EDF-2025 DA-SPACE-SBISR: Space-based ISR Funding Opportunity. EU Funding Portal. [Online]. Available: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/edf-2025-da-space-sbisr

[4] NASA Earth Science Division, Earth Science to Action Strategy, 2025. [Online]. Available: https://science.nasa.gov/earth-science/earth-science-to-action/

[5] N. Melega, N. Longepe, A Paskeviciute, et al, "Development and implementation of the Φsat-2 mission," Proc. SPIE 13546, Small Satellites Systems and Services Symposium (4S 2024), 135462X (20 March 2025); https://doi.org/10.1117/12.3062624

[6] D. Rijlaarsdam et al., "The Next Era for Earth Observation Spacecraft: An Overview of CogniSAT-6," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 18, pp. 2450-2463, 2025, doi: 10.1109/JSTARS.2024.3509734

[7] EU Council, Council conclusions on the use of satellite data, in particular in the space and security domains, ST-9288-2025-INIT, 2025.

[8] S. H. Miura, "Earth Observation data access interoperability implementation among space agencies," 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 2016, pp. 3621-3623, doi: 10.1109/IGARSS.2016.7729938

[9] ESA & CEOS, New Space Task Team White Paper – Overview and Recommendations, 2023. [Online]. Available: https://earth.esa.int

[10] World Economic Forum, Charting the Future of Earth Observation: Technology Innovation for Climate Intelligence, 2024. [Online]. Available: https://reports.weforum.org/docs/WEF_Charting_the_Future_of_Earth_Observation_2024.pdf

[11] EUROCONSULT, ST ENGINEERING. Edge computing in space: Unlocking value across satellite value chains. White Paper, 2023. Available: https://www.calameo.com/read/005503280e1c3da2978db

[12] NASA. Current Technology in Space – A Technology Taxonomy and Capability Snapshot. Briefing document, 2023. Available: https://nasa.epscorspo.neon-sandbox.org/media/uploaded_files/Current_Technology_in_Space_v4_Briefing.pdf

[13] P. Ghamisi et al., "Responsible Artificial Intelligence for Earth Observation: Achievable and realistic paths to serve the collective good," in IEEE Geoscience and Remote Sensing Magazine, doi: 10.1109/MGRS.2025.3529726

[14] ESA Phi-Lab, the University of Oxford, Trillium Technologies, “Earth System Predictability: How can AI Advance planetary stewardship?”, report from ESP forum, 2023; https://www.calameo.com/read/005503280e1c3da2978db

[15] French Institutes of Technology, “IRMA : un projet d’avant-garde pour l’observation de la Terre,” FIT BOOK, p. 37, 2025. [Online]. Available: https://www.french-institutes-technology.fr/wp-content/uploads/2025/04/L1106_FITBOOK25_Assemblage_PL.pdf

[16] A. Golkar and I. Lluch, “The federated satellite systems paradigm: Concept and business case evaluation,” Acta Astronautica, vol. 111, pp. 230–248, 2015.

[17] G. Denis, A. Claverie, X. Pasco, et al., “Towards disruptions in Earth observation? New Earth Observation systems and markets evolution: Possible scenarios and impacts,” Acta Astronautica, vol. 137, pp. 415–433, 2017.

[18] S. Tonetti, S. Cornara, G. V. de Miguel, et al., “Mission and system architecture for an operational network of earth observation satellite nodes,” Acta Astronautica, vol. 176, pp. 398–412, 2020.

[19] A. M. Wijata, M.-F. Foulon, Y. Bobichon, et al., “Taking artificial intelligence into space through objective selection of hyperspectral earth observation applications: To bring the ‘brain’ close to the ‘eyes’ of satellite missions,” IEEE Geoscience and Remote Sensing Magazine, vol. 11, no. 2, pp. 10–39, 2023.

[20] DOMINO-X Consortium, “DOMINO-X - Earth Observation Ground Segment for the Future,” [Online]. Available: https://domino-x.space/index.php/what-is-domino-x/

[21] DOMINO-E Consortium, “DOMINO-E - Your access to multi-mission Earth observation,” [Online]. Available: https://domino-e.eu/about/

[22] LEONSEGS Consortium, “LEONSEGS - Revolutionising the future of the Earth Observation data business landscape,”[Online]. Available: https://leonsegs.eu/about/

[23] CALLISTO Consortium, “CALLISTO: Copernicus Artificial Intelligence Services and data fusion with other distributed data sources and processing at the edge to support DIAS and HPC infrastructures,” [Online]. Available: https://callisto-h2020.eu/project/

[24] RapidAI4EO Consortium, “RapidAI4EO is advancing rapid and continuous land monitoring with state-of-the-art AI solutions,” [Online]. Available: https://rapidai4eo.eu/about/

[25] E. Kervennic, T. Louis, M. Benguigui, et al., “Embedded cloud segmentation using AI: Back on years of experiments in orbit on OPS-SAT,” in Proc. 2023 European Data Handling & Data Processing Conf. (EDHPC), pp. 1–8, 2023.

[26] T. Goudemant, B. Francesconi, M. Aubrun, et al., “Onboard anomaly detection for marine environmental protection,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 7918–7931, 2024.

[27] Eclipse Foundation, “Capella – Open Source MBSE Tool,” initially developed by Thales based on the Arcadia methodology. [Online]. Available: https://mbse-capella.org/

[28] Chell B., LeVine M. J., Capra L., Sellers J. J., Grogan P. T., New observing strategies testbed: A digital prototyping platform for distributed space missions, Systems Engineering, vol. 26, pp. 519–530, 2023. [Online]. Available: https://doi.org/10.1002/sys.21672

[29] J. Bonnet, M.-P. Gleizes, E. Kaddoum, et al., “Multi-satellite mission planning using a self-adaptive multi-agent system,” in Proc. 2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), pp. 11–20, 2015.

[30] B. Marchand, B. Francesconi, A. Girard, E. Kaddoum, and A. Perles, “Dynamic mission planning performances for agile Earth observing satellites with adaptive multi-agent system,” in Proc. International Workshop on Planning and Scheduling for Space (IWPSS), 2025.

[31] Z. Ge, S. Liu, F. Wang, et al., “YOLOX: Exceeding YOLO series in 2021,” arXiv preprint arXiv:2107.08430, 2021.

[32] T. Goudemant, B. Francesconi, H. Farhat, et al., “Détection de navires embarquable à bord de satellites,” in Proc. Conference on Artificial Intelligence for Defence (CAID), 2022.

[33]

Spiking Neural Networks for energy-efficient audio signals classification: representation matters

Noémie MARTIN
IRENav
École navale
Brest, France
martin53.eleve@ecole-navale.fr

Alban MAXIMIM
IRENav
École navale
Brest, France
maximin.eleve@ecole-navale.fr

Tristan AVERTY
IRENav
École Navale / Arts & Métiers ParisTech
Brest, France
tristan.averty@ecole-navale.fr

***Abstract*—Spiking Neural Networks (SNN) are a promising way of classifying time series, thanks to their energy efficiency and their ability to model biological temporal dynamics. The aim of this work is to study the influence of the form taken by the input data—1D raw signal vs. 2D time-frequency representation (spectrogram)—on the performance of a SNN in a binary classification task of sounds emitted by right whales. After searching for optimal hyperparameters using a 10-fold cross-validation, the results highlight that representing time series as spectrograms significantly improves time-frequency pattern discrimination and stabilizes network training, demonstrating the value of integrating a 2D representation for time series classification thanks to a SNN. These results are all the more interesting in that SNNs were originally introduced to handle one-dimensional time signals.**

***Index Terms*—Spiking neural network, binary classification, supervised learning, LIF neuron, signal processing, spectrogram.**

I. Introduction

Artificial neural networks (ANNs) such as convolutional neural networks (CNNs) or multilayer perceptrons (MLPs) have shown a great efficiency for numerous applications, among them data processing, object recognition and brain activity modelization [1]. Although designed for this purpose, these models remain only loosely inspired by the functioning of biological neurons and differ substantially from the temporal and energetic dynamics observed in the human brain [2]–[4].

In this context, a new class of so-called neuromorphic neural networks, spiking neural networks (SNNs), has been gaining increasing interest. Unlike ANNs, which use continuous functions to enable gradient backpropagation—an essential building block for training them—SNNs use spikes, adding a temporal dimension to data processing [1]. Nevertheless, SNNs remain globally complicated to train, mainly due to complex neuronal dynamics and the non-differentiable nature of neuronal discharge operations [2], [5].

Being inherently energy-efficient thanks to their event-driven impulse mode of operation, SNNs are seen as more energy-efficient than ANNs, as they replace energy-consuming weight multiplications with simpler additions [6] — making them particularly well suited to contexts where energy efficiency is crucial, such as embedded systems (e.g. UAVs, AUVs, autonomous cars) [2]. Their ability to process temporal information in real time [1], [6] also makes them relevant for onboard intelligence in operational environments, such as those encountered in naval or aerial systems. To fully exploit these advantages, SNNs must be implemented on neuromorphic hardware and coupled with event-based sensors [5], [7], [8], where computation occurs only upon spike events. Thus, their efficiency depends more on reducing firing spikes than on shrinking network size [6], allowing a new generation of energy-efficient and resilient AI systems.

As the underlying philosophy of SNNs is to model biological neural processes, they were initially developed for time series processing [8], in particular to save computational resources. Nevertheless, recent work in the literature shows the possible application of SNNs to image classification [9], thanks to hybrid architectures combining convolution layers with spiking neurons [2]. This dynamic is now being extended to more complex tasks, such as object detection, traditionally performed by architectures like YOLO (*You Only Look Once*), a convolutional neural network widely used for real-time object identification and localization. To this end, the "spiking-YOLO" model [10] has recently been proposed, combining the advantages of YOLO with the energy efficiency of SNNs, demonstrating the feasibility and relevance of SNNs in performance-constrained computer vision tasks.

These considerations position neuromorphic architectures as promising candidates for onboard signal analysis in unmanned maritime, aerial, or land systems. The detection of whale vocalizations, used here as a representative acoustic task, illustrates how such architectures can enable real-time and low-power signal processing in complex and uncertain environments, including passive acoustic detection, environmental awareness, or threat identification scenarios.

The aim of this study is to compare the performance of a spiking neural network in the context of a task involving the binary classification of audio recordings. The data used are 2-second time series, sampled at 2 000 Hz, containing or not right whale vocalizations, from a contest organized on the *Kaggle* platform. More specifically, this study aims to assess whether SNN performance is influenced by the nature of the input data: the raw signal (time series) or its spectrogram (image representation of the signal's time-frequency content).

The outline of this article is as follows: section II explains how a spectrogram is built and how an impulse neural network works, and in particular how the gradient is back-propagated in this very specific context. Section III then presents the database used and the preprocessing applied to it. Section IV discusses the different spiking neural network architectures tested. Section V describes and discusses the classification results obtained. Finally, a conclusion opens the way to a few perspectives.

II. Spiking neural network & spectrogram

A. Leaky Integrate-and-Fire (LIF) neuron

The *Leaky Integrate-and-Fire* (LIF) neuron model is one of the most common model in SNNs [6] mainly due to its simplicity and its low calculation costs [2], [4], [11]. It modelizes the membrane potential of a neuron that integrates inputs (excitatory or inhibitory) over time while gradually losing energy (hence the term *leaky*) to reach a resting value of the membrane potential, like an RC circuit [3] [12]:

$$\tau_m \frac{dU(t)}{dt} = -(U(t) - U_{\text{rest}}) + R_m I(t), \tag{1}$$

$$\tau_m = R_m C_m. \tag{2}$$

Equations (1) and (2) governs the LIF model. Equation (1) describes the temporal dynamic of the membrane potential $U(t)$ where

- τ_m is the membrane time constant (in seconds), defined by the relation (2) as the product of the membrane resistance R_m (in ohms) and the membrane capacitance C_m (in farads);
- $U(t)$ (in volts) is the electric potential at time t;
- U_{rest} (in volts) is the resting potential, towards which the membrane potential tends in the absence of stimulation;
- $I(t)$ (in amperes) represents the injected input current received by the neuron at time t;
- $R_m I(t)$ is the contribution of the input current to the membrane potential, translated into voltage by Ohm's law.

The equation (1) says that, in the absence of an injected current, the membrane potential decreases exponentially towards the resting potential with a time constant τ_m. When a current $I(t)$ is injected, it contributes to modifying the membrane potential in proportion to the membrane resistance. When a threshold is reached, the neuron emits a spike and the amplitude of the emitted spike is subtracted from the potential (1 in general) [13]. Figure 1 illustrates the evolution of membrane potential by aggregating input spikes. When this potential exceeds the threshold set at 0.5, an output spike is emitted.

This model therefore illustrates the essentials of neuronal dynamics, without however being as complex as detailed biophysical models such as Hodgkin-Huxley (HH) [3], [4]. The HH model improves upon the LIF model by providing a biophysically detailed description of how action potentials are generated through voltage-gated ion channels, offering a much closer match to real neuronal behavior [9]. However, this realism comes at a high computational cost. In contrast, the Izhikevich model offers a computationally efficient compromise between the LIF and HH models [8], [11].

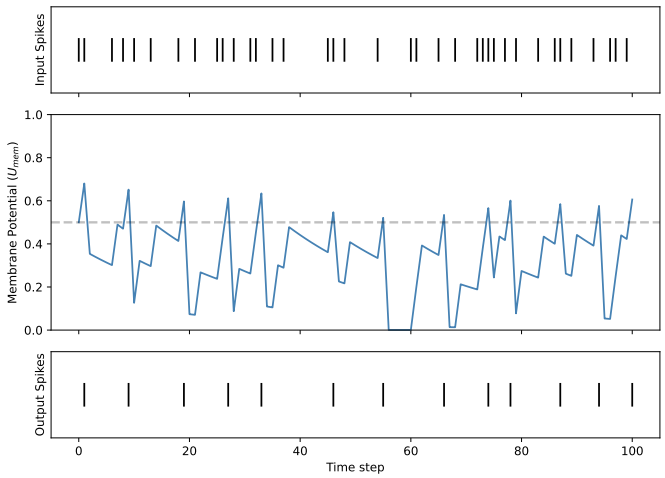


Figure 1. Evolution of membrane potential leading to the emission of 13 spikes.

B. Supervised learning

SNNs fall into the category of supervised learning algorithms and pose specific challenges due to the non-differentiable nature of the spikes, which are in a way binary activation functions [14]. To overcome this, the temporal backpropagation used for ANNs is adapted, allowing the error on the network weights to be backpropagated. The result is called surrogate gradient learning or spike-based backpropagation and replaces the non-derivable spike function with an approximate continuous function during learning [1] [2] [5]. These techniques make gradient descent possible in SNNs.

Let $\mathbf{y} = (y_i)_{1 \leq i \leq N}$ be the vector of N true labels ($y_i \in \{0, 1\}$ for all i) and $\hat{\mathbf{y}} = (\hat{y}_i)_{1 \leq i \leq N}$ be the vector of N probabilities predicted by the SNN ($\hat{y}_i \in [0, 1]$ for all i). The cost function used in supervised learning for a classification task is generally the cross-entropy :

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i). \tag{3}$$

As the spikes are non-differentiable functions, to apply optimization by gradient descent (*i.e.* backpropagation), the Dirac distribution $S(U)$ modeling a spike is replaced by a continuous function $\sigma(U)$, called surrogate, with a locally non-zero derivative (*e.g.* sigmoïd, bounded ReLU, ...) :

$$\frac{\partial S(U)}{\partial U} \approx \frac{\partial \sigma(U)}{\partial U}. \tag{4}$$

The approximate backpropagation of the gradient then becomes possible using the chain rule [6]

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial \sigma} \times \frac{\partial \sigma}{\partial U} \times \frac{\partial U}{\partial w} \tag{5}$$

where w is an arbitrary weight of the SNN.

The activation function plays a crucial role in a neural network, as it introduces the non-linearity required for learning. To make gradient backpropagation possible, the surrogate function σ used here is the variable-slope fast sigmoid function defined by

$$\sigma_s(x) = \frac{x}{1 + |sx|} \tag{6}$$

where $s > 0$ is an adjustable slope factor. Figure 2 illustrates the fast sigmoid function $\sigma_s(x)$ for different values of s, as well as its derivative. The greater the slope, the more the function resembles an impulse. However, this increase in slope is associated with an increasingly steep and localized derivative, which can make learning unstable. This trade-off between expressiveness and stability must therefore be carefully considered when choosing the slope [15].

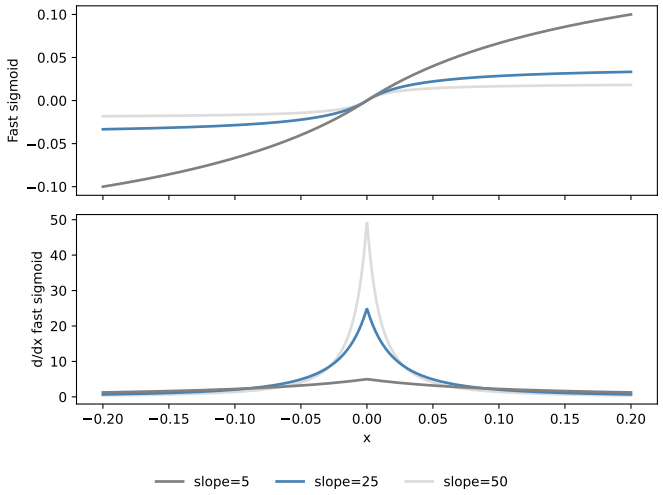


Figure 2. Fast sigmoid function and its derivative for different values of slope.

C. Spectrogram

In order to evaluate whether, in the context of a binary classification task, the performance of an SNN is influenced by the nature of the input data, we need to obtain a 2D representation of the audio signal. For this reason, we use in this article the spectrogram, the most widely used time-frequency representation. The spectrogram visualizes the evolution of a signal's frequency components over time. It is based on the Short-Time Fourier Transform (STFT), which consists of slicing the signal into short-duration segments using a sliding window that weights the signal to reduce the appearance of secondary lobes in the Fourier transform [16]. The Fourier transform is then applied to each of these segments. Formally, for a discrete signal $(x[n])_{0 \leq n \leq T-1}$ of T samples, the STFT is given by

$$X[n,k] = \sum_{m=0}^{T-1} x[m] \times h[n-m] \times \mathrm{e}^{-\mathrm{j}2\pi km/N_{\mathrm{fft}}}, \tag{7}$$

where $h[\cdot]$ is the analysis window (the Hamming window is used in this work), N_{fft} is the total number of points (*i.e.* the number of frequencies) of the Fourier transform and k is the frequential index. This transform calculates the spectral content of the signal around time n. Spectrogram resolution depends on several parameters : the number of frequencies considered N_{fft} determines the frequency resolution and for better temporal resolution, it is also possible to overlap the sliding windows by a number of points N_{overlap}.

III. Database & preprocessing

A. Database source and structure

The database used comes from the "Whale Detection Challenge" hosted on the Kaggle[1] platform. The data were provided by Marinexplore and Cornell University and consist of audio recordings captured via a network of buoys along the North American east coast. The data present a wide range of acoustic diversity, including anthropogenic (*e.g.* marine traffic) and biological noise, making the detection task particularly complex.

The database contains 30 000 two-second AIFF recordings, sampled at 2 000 Hz and annotated in a CSV file according to the presence (label 1) or absence (label 0) of right whale calls. The distribution is as follows: 76.6% of recordings are labeled 0 and 23.4% of recordings are labeled 1. By comparing a signal labeled 0 with one labeled 1, Figure 3 illustrates the complexity of the right whale call detection task.

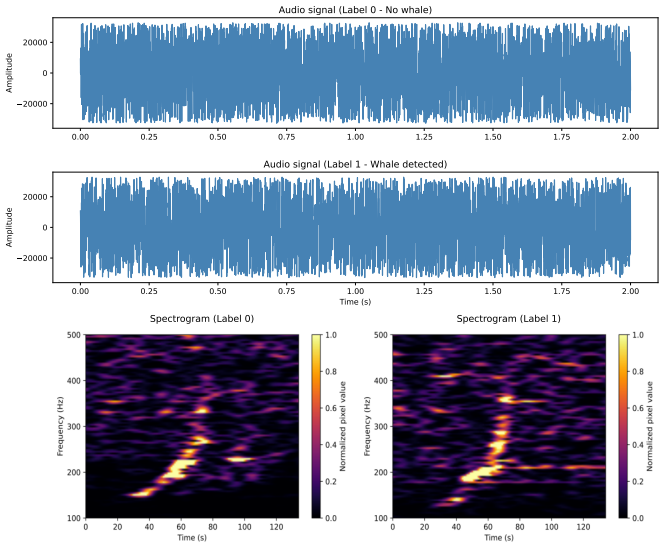


Figure 3. 1D (raw signal) and 2D (time-frequency) visualization of signals labeled 0 or 1.

B. Signal processing

Firstly, to improve learning, the database was processed in such a way as to balance the distribution between the two labels. Hence, 23.4% of the signals that are labeled 0 are randomly selected. It corresponds to 7027 time series.

[1] https://www.kaggle.com/c/whale-detection-challenge

The processing applied to the raw audio data begins with 4$^{\text{th}}$ order bandpass filtering to retain only the frequency components between 100 Hz and 500 Hz (a choice motivated by biological reasons). Next, a threshold at 1 and 99$^{\text{th}}$ percentiles is applied to limit the impact of extreme values. The signal is then decomposed into two distinct channels: one containing positive values, the other negative values. These two vectors are concatenated and the absolute values are taken to obtain a positive representation of the signal. This transformation is essential, as SNNs require inputs between 0 and 1. A simple normalization between 0 and 1 would distort the energy dynamics of the signal by assigning an average energy around 0.5 even to a silent signal, leading to saturation of the network with spikes. Therefore, this method preserves the energy coherence of the original signal, while partially preserving the phase information. Each signal is processed individually, taking into account its own extreme values. Figure 4 depicts the above-mentioned processing on a synthetic time series.

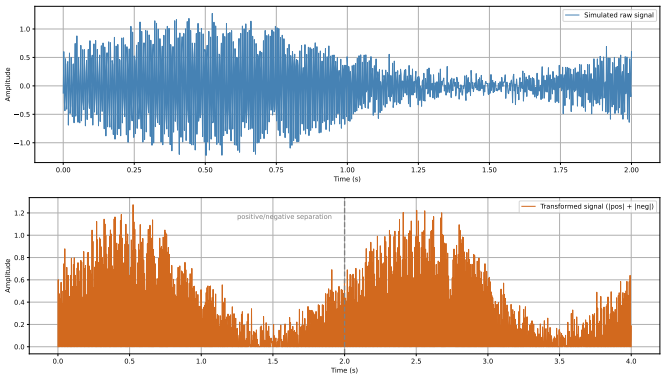


Figure 4. Up : Raw synthetic time series / Down : Preprocessed synthetic time series (separation and concatenation of components)

The processing applied to the spectrograms also includes a thresholding step at the 99$^{\text{th}}$ percentile. This is followed by filtering of frequencies between 100 Hz and 500 Hz, then normalization between 0 and 1. Each spectrogram is processed individually, taking into account its own extreme values.

IV. Methodology

A. Tools & libraries

The general construction of the neural networks is implemented using the `pytorch` library. To integrate the spiking character specific to this study's approach, we used `snntorch`, a library specially designed to be used as an extension of `pytorch` for spiking neural networks. It provides adapted modules (spiking neurons, time-coding mechanisms, etc.) to simulate the discrete-time behavior of this type of neuron. The `scikit-learn` library is used for performance evaluation, notably using the AUC-ROC metric, as well as for cross-validation. Finally, some processing operations require signal operations (filtering, spectrogram generation), which are performed with the `scipy` module.

B. Spiking neural network architecture

Table I summarizes the architectures of the two considered networks. The encoding of the data in spikes is made by the input layers. The static data is thus treated as a direct current (DC), whose characteristics are transmitted to the first layer of the network at each time step [3]. The neuron model chosen is the LIF neuron as presented in subsection II-A.

	1D input	**2D input**
Input layer	8 000	14 175 (105×135)
Hidden layer 1	2 048	4 096
Hidden layer 2	512	512
Hidden layer 3	64	64
Output layer	2	2

Table I
NUMBER OF NEURONS PER LAYER FOR THE TWO NETWORKS.

C. Hyperparameters optimization

Firstly, a K-fold cross-validation is performed to train the networks and find the best hyperparameters. As a reminder, cross-validation is a statistical evaluation method used to measure a model's ability to generalize and thus optimize hyperparameters while limiting overfitting. Formally, the database, composed of 14 054 elements, is divided into K subsets and the model is trained K times: at each time, $K-1$ subsets are used to train the neural network and the K^{th} is used for validation. Finally, the scores are averaged over the K validations to obtain the global metric. The following hyperparameters are set:

- Batch size : 512
- Spike emission threshold : 1
- Initial learning rate : 10^{-5}
- Slope of the surrogate fast sigmoïd function : 25

The use of a scheduler such as `ReduceLROnPlateau` for training allows to dynamically adjust the learning rate (an essential parameter of a gradient descent) in response to performance stagnation on the validation set, measured via the loss function. Given the complexity of optimization in SNNs, this adaptation stabilizes learning, facilitates convergence and achieves better performance while reducing the risk of overfitting. Concretely, the patience of the scheduler corresponds to the number of epochs where the value of the loss function can stagnate (*i.e.* vary by a value below the set threshold) before the learning step decreases proportionally to the decrease factor. Thus, the list of set hyperparameters is completed by those of the scheduler:

- `patience` : 5 for the SNN with 1D inputs and 3 for the SNN with 2D inputs
- `factor` : 0.5
- `threshold` : 10^{-4}

Finally, a grid-search is performed to find the best combination of hyperparameters `beta` (leakage rate of LIF neurons) and `num_step` (number of time steps):

- `num_step` : search among $\{1, 10, 20, 30, 40, 50\}$
- `beta` : search among $\{0.8, 0.9, 0.98, \textit{learnable}\}$

The *learnable* option means that the `beta` parameter is initially set to 0.9 and then optimized during training as a network parameter.

V. Results & discussions

A. Performance score

The ROC curve (for Receiver Operating Characteristic) is a curve used to evaluate the performance of a binary classifier. It consists of plotting the true positive rate against the false positive rate for different threshold values, the threshold being set to decide whether the neuron output predicts 1 or 0. Calculating the area under the curve (AUC) summarizes the quality of the neural network across all thresholds [17]. The AUC is particularly interesting because it is independent of class distribution and the arbitrary choice of a classification threshold. What's more, unlike a simple accuracy measure, it reflects the model's ability to maximize true positives while minimizing false positives, which is crucial in sensitive applications where the cost of errors differs according to their nature [18].

B. Results

To compare the different configurations tested, the median of the AUC score on the K-folds was preferred to the mean. This is because the SNN model has an instability intrinsic to training that can lead to learning failures on certain folds, as illustrated in Figure 5. In this context, the mean is sensitive to these extreme values, while the median provides a more robust and representative estimate of actual performance.

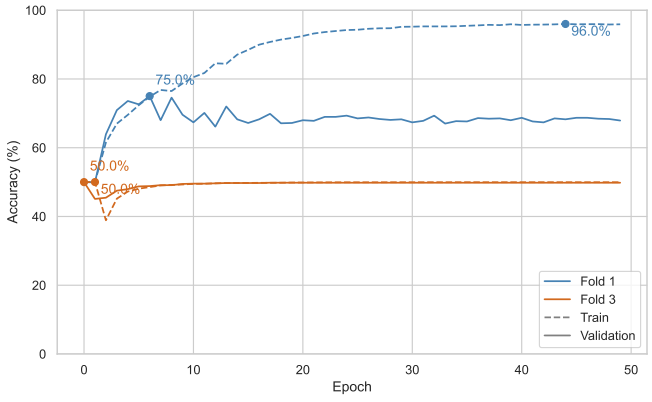


Figure 5. Illustration of learning failure on fold 3 when training the network with 1D inputs with identical hyperparameters (`beta` $= 0.7$ and `num_step` $= 40$).

Tables II and III summarize the median AUC scores obtained on the different cross-validation datasets according to the values of parameters `beta` and `num_step`. The best results are highlighted.

beta \ num_step	1	10	20	30	40	50
0.7	0.725	0.746	0.736	0.751	0.759	0.746
0.8	0.726	0.735	0.745	0.738	0.740	0.747
0.9	0.716	0.731	0.729	0.728	0.742	0.742
0.95	0.724	0.728	0.715	0.718	0.740	0.712
learnable	0.727	0.731	0.737	0.738	0.747	0.732

Table II
Median AUC scores according to `beta` and `num_step` parameters for the network with 1D inputs.

beta \ num_step	1	10	20	30	40	50
0.7	0.737	0.920	0.922	0.921	0.911	0.891
0.8	0.648	0.922	0.921	0.925	0.923	0.919
0.9	0.502	0.923	0.925	0.925	0.927	0.918
0.95	0.602	0.922	0.925	0.926	0.926	0.926
learnable	0.500	0.924	0.924	0.926	0.927	0.923

Table III
Median AUC scores according to `beta` and `num_step` parameters for the network with 2D inputs.

C. Discussion

Performance evaluation was based on the best median AUC scores obtained from K-fold cross-validation, for each architecture tested. The best SNN model receiving raw 1D data (temporal audio signal) achieved a median score of 0.759 with the optimal hyperparameters `beta` $= 0.7$ and `num_step` $= 40$. In comparison, the best SNN model taking 2D inputs (spectrograms) achieves a significantly higher score of 0.927 for `beta` $= 0.9$ and `num_step` $= 40$. These results clearly show that the representation of input data has a major impact on the performance of an SNN. Exploiting the time-frequency structure of the signal through 2D spectrograms enables the network to extract discriminating patterns more efficiently, particularly in an audio classification context. Conversely, 1D raw data appear to be insufficiently informative for efficient learning with an SNN.

Beyond performance, learning dynamics reveal clear differences between the two approaches. The network which takes 1D inputs shows a tendency towards overfitting: validation accuracy peaks before decreasing slightly, while training accuracy continues to rise. Learning is also unstable, with two out of ten folds showing no progression (the *accuracy* stagnate around 50%), indicating a lack of convergence. In comparison, the network which takes 2D inputs, shows stable and regular learning: the accuracy curves for training and validation follow parallel trajectories, with no observable overfitting. This stability is consistent both within and between folds. These results confirm that spectrogram (2D input) is better suited to SNNs, offering both better performance and more reliable learning for audio signal classification.

Finally, it should be noted that the test dataset was only provided to contest participants, so we are unable to accurately compare ourselves with their models, the results of which are published on the contest website[2] and, in some cases,

[2]https://www.kaggle.com/c/whale-detection-challenge/leaderboard

have been published [19], [20]. However, we must remain realistic: the results obtained using their methods will certainly be much better than those obtained with our "simple" SNN architectures. Nevertheless, the real objective of this work was to focus specifically on the influence of representations on performance and training behavior.

VI. Conclusion

This work confirms that, for the binary classification of audio signals by an SNN, a 2D time-frequency input (spectrogram) provides both better performance and higher learning stability than the 1D raw signal. Switching to a time-frequency representation enables discriminative features to be extracted more efficiently, leading to an AUC increase of over 0.15. The performance obtained on the train set can be complemented by results obtained on a test set.

Although this representation is naturally two-dimensional, it is vectorized before being injected into the SNN, and thus treated as a 1D vector. This means that no convolution operation is currently used to explicitly exploit the local spatial correlations present in the spectrograms. This limitation raises a potential improvement: the integration of spatio-temporal convolutional layers could improve network performance while remaining compatible with the constraints of deployment on neuromorphic hardware.

In parallel, sample-wise SNNs have been developed and tested, capable of processing each audio sample step by step, as well as neuromorphic recurrent loop architectures to better capture fine temporal dynamics. These approaches, combined with vectorized 2D representation, open up promising perspectives for enhancing both the robustness and energy efficiency of SNNs for audio signal classification tasks. Such properties—real-time operation, low energy consumption, and temporal adaptability [21]—make these architectures particularly relevant for future embedded applications requiring autonomous and resilient perception in complex environments. Moreover, it should be noted that some recent studies show that it might be more beneficial to use hybrid ANN (for parallel processing) / SNN (for event-based approach) architectures for further energy savings [22].

References

[1] Guillaume Marthe. *Neurones à impulsion pour les communications sans fil*. PhD thesis, Institut National des Sciences Appliquées de Lyon, 2024.

[2] Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking Neural Networks and Their Applications: A Review. *Brain Sciences*, 12(7):863, 2022.

[3] Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. Training Spiking Neural Networks Using Lessons From Deep Learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

[4] Veïs Oudjail. *Réseaux de neurones impulsionnels appliqués à la vision par ordinateur*. PhD thesis, Université de Lille, 2022.

[5] João D. Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S. Cardoso. Spiking neural networks: A survey. *IEEE Access*, 10:60738–60764, 2022.

[6] Yufei Guo, Xuhui Huang, and Zhe Ma. Direct Learning-Based Deep Spiking Neural Networks: A Review. *Frontiers in Neuroscience*, 17:1209795, 2023.

[7] Michael Pfeiffer and Thomas Pfeil. Deep Learning With Spiking Neurons: Opportunities and Challenges. *Frontiers in Neuroscience*, 12:774, 2018.

[8] Hagar Hendy and Cory Merkel. Review of spike-based neuromorphic computing for brain-inspired vision: biology, algorithms, and hardware. *Journal of Electronic Imaging*, 31, 2022.

[9] Y. Dan, Z. Wang, H. Li, and J. Wei. Sa-snn: Spiking attention neural network for image classification. *PeerJ Computer Science*, 10, 2024.

[10] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11270–11277, 2020.

[11] Yin Bojian. *Efficient and Accurate Spiking Neural Networks*. PhD thesis, Eindhoven University of Technology, 2022.

[12] Jiankun Chen, Xiaolan Qiu, Chibiao Ding, and Yirong Wu. SAR Image Classification Based on Spiking Neural Network through Spike-Time Dependent Plasticity and Gradient Descent. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188:109–124, 2022.

[13] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[14] Hyeryung Jang, Osvaldo Simeone, Brian Gardner, and Andre Gruning. An introduction to probabilistic spiking neural networks: Probabilistic models, learning rules, and applications. *IEEE Signal Processing Magazine*, 36(6):64–77, 2019.

[15] Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks. *Neural Computation*, 30(6):1514–1541, 2018.

[16] A. O. Boudraa. Traitement du signal avancé. https://sites.google.com/view/aboudra/teaching, 2024. Accessed in may 2025.

[17] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[18] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] Evgeny Smirnov. North atlantic right whale call detection with convolutional neural networks. In *Proc. Int. Conf. on Machine Learning, Atlanta, USA. Citeseer*, pages 78–79, 2013.

[20] Kostiantyn Pylypenko. Right whale detection using artificial neural network and principal component analysis. In *2015 IEEE 35th International Conference on Electronics and Nanotechnology (ELNANO)*, pages 370–373. IEEE, 2015.

[21] Sales G Aribe Jr. Spiking Neural Networks: The Future of Brain-Inspired Computing. *arXiv preprint arXiv:2510.27379*, 2025.

[22] Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. Are SNNs really more energy-efficient than ANNs? An in-depth hardware-aware study. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):731–741, 2022.

FedSegMIA: Exploring Privacy Risks in Federated Binary Segmentation

Eugénie Laugier
Thales
cortAIx Labs, France

Alice Héliou
Thales
cortAIx Labs, France

Vincent Thouvenot
Thales
cortAIx Labs, France

Katarzyna Kapusta
Thales
cortAIx Labs, France

alice.heliou@thalesgroup.com vincent.thouvenot@thalesgroup.com katarzyna.kapusta@thalesgroup.com

***Abstract*—Federated learning (FL) enables collaborative training of machine learning models across multiple parties without sharing raw data, making it particularly appealing for defense applications involving sensitive or classified information. While the privacy risks of FL have been extensively studied for classification tasks, vulnerabilities in federated segmentation models, which are widely used for precise object detection and reconnaissance, remain largely unexplored. Existing studies on segmentation have focused on centralized settings, typically relying on prediction losses as the main leakage vector.**

In this work, we present the first systematic analysis of membership inference attacks on binary segmentation models trained under FL. We demonstrate that gradient updates provide a significantly stronger signal for inferring training data membership than losses, posing substantial risks in collaborative defense scenarios. Our experiments highlight the need of implementing robust privacy-preserving mechanisms to protect critical operational data.

***Index Terms*—membership inference attacks, automatic target detection, segmentation, federated learning, privacy**

I. Introduction

In modern military operations, multiple units (from ground vehicles to reconnaissance drones) must collaboratively build a shared understanding of the battlefield, without exposing sensitive imagery. Collaborative learning offers a solution by enabling each system to improve its perception capabilities while keeping raw data private. To be effective, these systems must detect, recognize, and precisely locate objects to operate in complex environments. This makes semantic segmentation a model of choice, as it provides precise pixel-level classification, essential for identifying targets and distinguishing allies from adversaries. But, this setup raises a critical question: could collaborative learning of semantic segmentation models unintentionally leak sensitive information ?

Federated learning (FL) [1] allows multiple parties to collaboratively train a machine learning model without sharing their raw data. The idea is that each party has its local data and only model updates on the local data are shared with an aggregation server that orchestrates the training. By enabling multiple clients to jointly optimize a global model while keeping their data local, FL offers an attractive solution for privacy-preserving learning in critical domains such as healthcare [2], [3], finance [4], and increasingly, defense [5].

Although federated learning is designed to reduce privacy risks, the information exchanged during training can still leak sensitive details. A growing line of work has investigated privacy attacks that exploit these shared model updates. Among them, membership inference attacks (MIAs) [6] are particularly concerning. As illustrated by Figure 1, MIAs seek to determine whether a specific data point was part of a client's training set, exploiting the tendency of machine learning models to memorize training data. In the context of MIA, a 'member' refers to data that is part of the training dataset of the targeted model, while a 'non-member' denotes data that is not included in this dataset. Their principle relies on the fact that the model behaves differently between training data and unseen data. Unlike gradient inversion attacks (GIAs) [7], which attempt to reconstruct input data from shared gradients and often rely on restrictive conditions such as small batch sizes or minimal local training epochs, MIAs can operate under more realistic assumptions. Consequently, MIAs pose a practical and significant threat in federated environments.

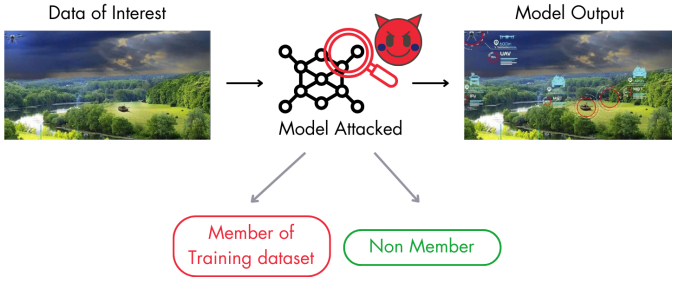


Fig. 1. MIA principle in centralized learning.

While research has focused on MIAs in the context of classification tasks [8]–[10], significantly less attention has been paid to other machine learning applications that also handle highly sensitive data. In particular, semantic segmentation has received little scrutiny regarding its vulnerability to membership inference in federated learning. This gap is especially concerning given the growing interest in deploying segmentation models on edge devices such as drones [11], [12]. These devices increasingly rely on federated learning to collaboratively improve perception models without transmitting raw imagery back to a central server.

In this work, we address this gap by examining server-side membership inference attacks during federated learning

of binary segmentation models.

- We evaluate the effectiveness of membership inference attacks on federated binary segmentation, highlighting that even without strong assumptions about attacker's capabilities, these models are susceptible to privacy breaches.
- We improve over the state-of-the-art by showing that artificially increasing the number of clients can introduce a bias that amplifies the effectiveness of gradient-based attacks.
- Finally, we hypothesize that certain iteration rounds exhibit stronger susceptibility to inference attacks than others. By leveraging the segmentation performance of the federated model to identify and select these more informative iterations, we demonstrate that the server can enhance the effectiveness of inference .

Our findings underscore the need of addressing privacy vulnerabilities and server-side threats in federated learning, especially as it is applied to complex tasks like segmentation.

II. Business Need / Motivations

Federated learning is inherently suited for scenarios involving sensitive or confidential data, making it an attractive approach across domains such as healthcare, finance, and particularly defense. In military contexts, FL can be deployed to collaboratively trained models across multiple entities without sharing raw data, preserving operational secrecy. Beyond centralized installations, federated learning holds promise for integration directly on edge platforms such as autonomous vehicles and drones; enabling these systems to continuously improve object detection or segmentation capabilities by learning from local observations collected in diverse environments. For reconnaissance drones in particular, this means adapting models in real time to new terrains or targets, without ever transmitting potentially sensitive imagery back to a central server. Additionally, by sharing only model updates instead of raw data, FL also helps lower the communication costs associated with centralized learning.

While reconstructing complete images from model updates remains a technical challenge, subtle forms of information leakage, such as revealing data characteristics (e.g., image resolution or content type) or identifying which clients participated in training, could still pose serious risks. This underscores the importance of thoroughly understanding privacy vulnerabilities in federated learning.

III. Related Work

A. Federated Learning

Existing FL architectures can be either centralized (Fig.2), relying on a server to aggregate local updates, or fully decentralized, where clients communicate peer-to-peer [13]. In centralized settings, the server coordinates the learning process [14], and at each iteration it collects all local updates and aggregates them. This aggregation process grants the server access to a substantial amount of information, positioning it as a potentially powerful adversary. In our work, we will only focus on a centralized federated setting and demonstrate the extent to which a server can infer information about local datasets. However, recent research has shown that significant privacy risks persist even in decentralized settings [15].

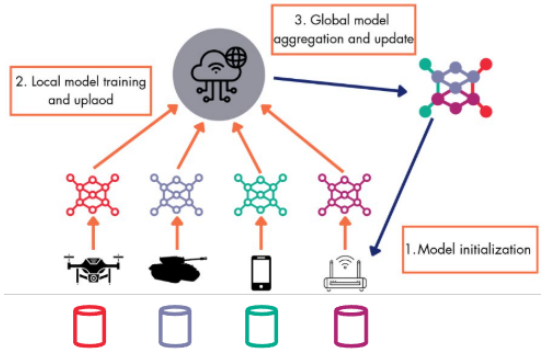


Fig. 2. Principle of Centralized Federated Learning.

B. MIA in a Federated Setting

MIAs leverage the observation that machine learning models tend to behave differently on data they have seen during training versus unseen data: producing lower losses or more confident predictions on training samples [8]. Various strategies have been proposed to exploit this discrepancy, differing mainly in the metrics they use and the level of adversary involvement, from passive observation to actively training shadow models.

1) Attacks based on model updates: A significant body of work investigates MIAs that exploit information contained in model updates exchanged during federated learning.

Gradient-based attacks compare raw gradients, their norms, or compute metrics such as cosine similarity to distinguish members from non-members [9], [16], [17]. These methods are often highly effective in classification and entirely passive, but require direct access to model gradients (white-box scenario).

Loss-based attacks instead rely on the observation that the loss is typically lower for training data [18]. These attacks can be carried out without access to the model architecture and weights (black-box settings).

Another classical strategy involves shadow training, where the attacker builds one or more shadow models on data drawn from a distribution similar to that of the target. These shadow models are then used to train membership classifiers that predict whether a given sample was part of the target's training set [19] [20]. While this approach is often more precise, it requires substantial auxiliary data and considerable computational resources, making it significantly more expensive than above attacks, which typically only involve running inference or computing gradients on the target model.

More intrusive attacks involve manipulating local models or the training process itself to introduce vulnerabilities that can later be exploited [21], [22]. Such methods reduce the need for data but are more detectable by traditional defense methods.

2) Attacks based on training dynamics: Other techniques analyze how certain metrics evolve over multiple training

rounds, typically without requiring access to labels or explicit gradients. This places them in a largely black-box setting, imposing fewer constraints on the attacker. For example, some approaches monitor the evolution of the loss across federated iterations [23]–[25], while others track how prediction confidences change over time [16], [26]. More recent methods examine shifts in the bias terms of the final layer [27]. However, these strategies are often more sensitive to training dynamics, and some still rely on access to prediction confidences or internal parameters, which may not always be available in practice.

C. FedMIA [10]

Instead of training shadow models to obtain additional information at a significant computational costs, [10] proposes an approach that leverages information from non-target clients. Assuming that clients' datasets are disjoint, they demonstrate that it is possible to estimate the distribution of attack signals (such as losses or gradients) for models that are not trained on the target data (the "non member" distribution). Then, by employing a one-tailed likelihood-ratio hypothesis test using the estimated non-member distribution, they can infer whether the target data was part of the training dataset for the targeted client. The Figure 3 illustrates the FedMIA approach. Our work builds in part on this method by combining it with shadow models to further enhance the information accessible to the central server.

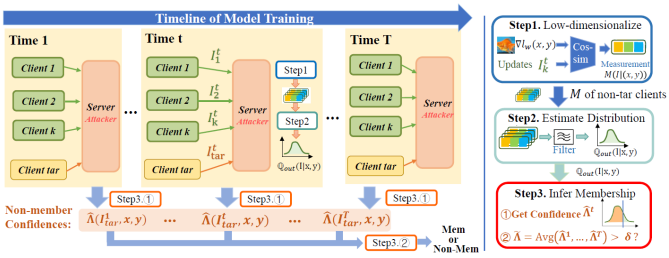


Fig. 3. All vs Target Attack Principle: FedMIA [10] Overview.

D. MIA on Segmentation Models

Segmentation tasks fundamentally differs from classification by predicting a label for each pixel rather than assigning a single label per image, and typically employ pixel-wise losses that compare entire spatial maps.

The richer output structure of segmentation models may affect both the type of privacy leakage and the metrics that are most informative for inference attacks.

Although some studies have investigated MIAs against segmentation models under conventional centralized training, this body of work remains relatively limited. Most approaches exploit the segmentation loss, operating under the hypothesis that it reveals more about individual data samples than in standard classification settings. Early work on membership inference for segmentation models focused on exploiting localized loss signals. He et al. [28] proposed a patch-based analysis of the loss map, showing that certain spatial regions

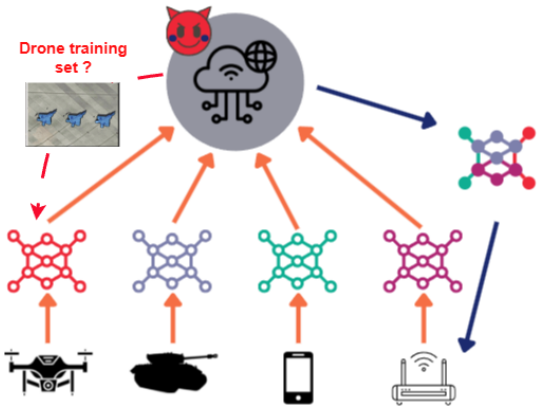


Fig. 4. MIA Threat Model in Federated Learning.

of the output can carry stronger membership signals than global loss alone. Building on this direction, Chobola et al. [29] conducted a more comprehensive study, distinguishing between binary and multiclass segmentation settings. Using shadow models, they evaluated three attack strategies: one based on global loss, one using patch-wise loss, and another combining the target model's predictions with the ground truth masks. Their findings revealed that, somewhat surprisingly, the global loss often remained the most effective signal. However, all these investigations have been confined to centralized learning, leaving open the question of how segmentation models trained under federated learning might be vulnerable to membership inference. In this work, we address this gap by presenting the first systematic study of MIAs on federated binary segmentation models, while also evaluating signals beyond the commonly examined loss.

IV. Membership Inference Attack on a Binary Segmentation Model in a Federated Setting

A. Threat Model

We consider a federated learning setting with a centralized architecture, where a server orchestrates the collaborative training of a global model by aggregating updates received from multiple clients. In this context, we examine the scenario in which the server aims to infer private information from the clients' contributions, without seeking to interfere with the federated learning process.

Figure 4 provides an illustration of this threat model.

The server has direct access to all model updates exchanged during training, including their weights, architecture and hyper-parameters, corresponding to a white-box scenario. We consider two levels of attacker behavior.

- Passive scenario: the server respects the FL protocol without interfering with client operations. Its only intervention consists in executing additional inference passes on candidate samples.
- Proactive scenario: the server injects artificial clients into the training process to simulate non-member behaviors. This allows the server to better characterize non-member

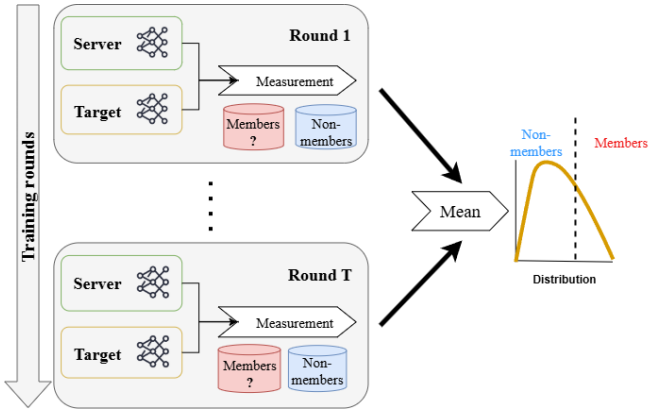


Fig. 5. Server vs Target Attack Principle.

samples and thus more easily distinguish them from members.

In both cases, we assume all clients behave honestly, they do not collude with the server or with each other and do not try to poison the training.

B. Notations

We consider a federated learning process over T communication rounds with K clients. Let (x, y) denote a target data instance, where x is an input image and y its corresponding ground-truth segmentation mask. We use $W_S^{(t)}$ to represent the global model weights held by the server at the end of round t, and $W_S^{(t-1)}$ for the previous round. Similarly, $W_k^{(t)}$ denotes the local model weights trained by client k during round t. The local update sent by client k is given by Equation (1).

$$I_k^t = W_k^{(t)} - W_S^{(t-1)}. \tag{1}$$

We further denote by $\nabla_W L(y, x; W_S^{(t)})$ the gradient of the loss function L evaluated on (x, y) with respect to the global model parameters. This quantity captures the direction in parameter space that would most improve prediction on the specific instance (x, y).

C. Type of MIAs

In this work, we investigate two main dimensions of membership inference attacks on federated segmentation models: the attacker's comparison strategy and the signal exploited to infer membership.

1) Comparing Server-Only and All-for-One Attacks: We explore two different attacker perspectives based on how the server leverages the information available from local updates.

- **Server vs Target:** In this classical approach, the server focuses exclusively on the update received from the targeted client. As illustrated by Fig.5, at each communication round t, for a given candidate instance (x, y), the server computes an indicator of membership (such as cosine similarity or loss difference) by comparing the targeted client's update $I_k^t = W_k^{(t)} - W_S^{(t-1)}$ to the behavior of the global model.
- **All vs Target:** This more recent approach, inspired by FedMIA [10], leverages the updates from all participating clients. For each round, the server estimates the distribution of membership signals across non-target clients, treating them as a baseline under the null hypothesis that they did not train on (x, y). A statistical test is then performed to assess whether the target client's signal significantly deviates from this distribution, thus providing a confidence measure (p-value) for inferring membership.

2) Attack Signal: We evaluate two primary signals used to distinguish member from non-member data:

- **Cosine Similarity of Gradients:** For each candidate instance, we compute the gradient of the loss with respect to the global model parameters at round t, denoted $\nabla_W L(y, x; W_S^{(t)})$. The cosine similarity between this gradient and the targeted client's update measures their alignment:

$$\text{cosim}(x, y) = \frac{\langle \nabla_W L(y, x; W_S^{(t)}), I_k^t \rangle}{\|\nabla_W L(y, x; W_S^{(t)})\|_2 \cdot \|I_k^t\|_2}. \tag{2}$$

  Empirically, gradients associated with independent data tend to be nearly orthogonal in high-dimensional spaces, so a significantly higher similarity indicates that (x, y) may have been used to train the local model.
- **Loss Difference:** This simpler attack compares the loss values computed by the global model and by the target client. In segmentation, the pixel-wise nature of the loss function provides a finer-grained signal than typical classification settings. The hypothesis is that if (x, y) was seen during local training, the discrepancy in loss between the server and the client's update will be statistically smaller.

V. Experimental Setup

A. Dataset and Model

We restrict our study to binary segmentation, both for simplicity and as a first step toward understanding membership inference vulnerabilities in federated segmentation models. Our target model is a UNet [30], a widely used architecture in segmentation tasks. We employed the iSAID dataset [31], which contains aerial images from complex scenes annotated across 15 object classes, including ships, aircraft, and harbors. This dataset was chosen as it most closely resembles the Automatic Target Detection/Recognition use case, which is particularly relevant for application of segmentation models in defense, with relevant classes and varying image resolutions. To adapt it to a binary segmentation task, we filtered the dataset to retain only images containing at least one instance of type *harbor*. After filtering, the final dataset comprised 311 images, of which 281 were used for training and 30 for testing. Figure 6 shows an example image and its corresponding mask.

B. Data Distributions

Given the limited number of training images, we explored two data distribution strategies, each aligned with one of the attacker behaviors introduced in Section IV-A.

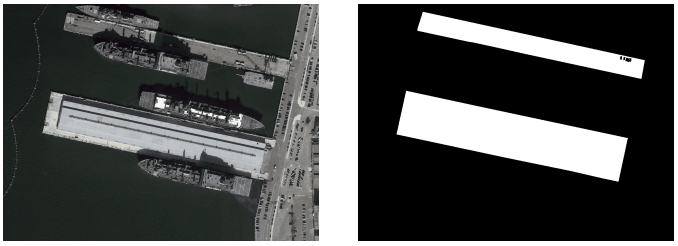


Fig. 6. Image (left) and Label (right) of the filtered iSAID [31] dataset.

For the first scenario, training data was distributed equally among clients. We limited experiments to 3 and 5 clients, as beyond that threshold, each client receives too few images, making local training less meaningful.

To simulate the proactive server scenario, we adopted an overlapping data distribution. In this setup :

- The clients 1 (the target) and 2 receive 25% of the training data (70 images each),
- The remaining 50% of the training data (called the shadow dataset) is randomly split among synthetic clients, such that all clients hold 70 images. We use a draw with replacement, so that an observation can be simultaneously be present in two (or more) clients. This allows us to simulate a larger number of clients without reducing the data available to the target client.

We conducted experiments with 5, 10, and 20 clients under this setting. To construct evaluation datasets, we sampled member instances from the target client's training data, and non-member instances from the concatenation of the test set and the training data from other clients.

C. Training Setup

We trained the global model using the classical FedAvg [14] aggregation scheme, computing a simple average of updated weights from all clients. All images were resized to 300×350 pixels. Training employed the Adam optimizer with learning rates ranging from 10^{-5} to 10^{-2}. To mitigate class imbalance in the masks, we used a weighted binary cross-entropy loss defined by:

$$L_{\text{balanced}}(y, \hat{y}) = L(y, \hat{y}) \times (1 - \alpha + \alpha\, y_{\text{balanced}}),$$

where L is the standard binary cross-entropy, $\hat{y}$ the predicted mask, and y_{balanced} is calculated as

$$y_{\text{balanced},i,j} = \begin{cases} \gamma & \text{if } y_{i,j} = 0 \\ 1 - \gamma & \text{otherwise} \end{cases}$$

with γ equal to the proportion of positive pixels in y.

We investigated the influence of the number of local epochs per client by testing values of 1, 2, and 4. We kept the total local training epochs fixed at 100, resulting in 100, 50, or 25 communication rounds respectively. This ensures equivalent data exposure across all configurations, which is essential for future studies incorporating differential privacy. We hypothesized that increasing local epochs could amplify attack success by widening the gap between local and global models.

All experiments used a fixed random seed. Due to computational constraints, variability was evaluated on a single configuration (1 local epoch, 100 rounds, learning rate 10^{-3}), repeated ten times.

D. Evaluation Metrics

We evaluated segmentation model performance using the DICE coefficient, defined as

$$\text{DICE} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN},$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. In our binary segmentation setting, white pixels represent True values, and black pixels False. The DICE score therefore measures the degree of overlap between the predicted and ground-truth masks.

For attacks, we followed standard membership inference metrics, primarily reporting the area under the ROC curve (AUC) and the true positive rate at a fixed false positive rate (TPR@FPR), with FPR set to 0.01. The AUC provides a global assessment of an attack's performance across all possible decision thresholds, while TPR@FPR focuses on a critical operating point where false positives must be tightly controlled. This is particularly relevant for sensitive applications that require stringent control over false positives. We compute the membership inference metrics using the training dataset of the target client as 'member'. Then we either use the test dataset alone as 'non-member' or the concatenation of the test dataset with the other clients dataset. Unless stated otherwise, the later metric is used on the presented results.

VI. Experimental Results

Unless stated otherwise, all results presented come from experiments conducted using a learning rate of 10^{-3} with one local epoch per client.

A. Passive Attacker Scenario

Table I present the results obtained under the passive server setting, where data was distributed equally among clients. Due to the limited dataset size, this configuration could only be evaluated with 3 and 5 clients, reflecting the scenario where for instance several organizations collaborate together in the framework of a military mission involving surveillance drones.

In this setting, FedMIA Loss achieves the best result with an AUC of 65.5% and TPR@FPR of 8.9% at 5 clients. With only 3 clients, all attacks perform poorly, as the reduced number of client limits the attacker's ability to extract meaningful signals. Moreover, FedMIA's statistical test is theoretically valid only with at least 5 clients, though we included the 3-clients case for empirical completeness.

These observations suggest that in a setting featuring few clients and limited overfitting, membership inference attacks remain relatively ineffective. This motivates the investigation of a proactive server strategy, where the server can inject artificial clients to strengthen its ability to model non-member behavior statistically.

TABLE I
AVERAGE ATTACKS RESULTS IN THE PASSIVE SCENARIO BY CLIENT COUNT (100 ITERATIONS, LEARNING RATE 10^{-3}).

Data Distribution	N° Clients	Cosine Similarty		Loss Difference		FedMIA Cosine		FedMIA Loss	
		AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01
Normal	3	55.6%	2.2%	58.3%	2.2%	60.0%	4.3%	**61.8%**	**6.5%**
	5	57.2%	1.8%	62.4%	5.4%	61.8%	7.1%	**65.5%**	**8.9%**

TABLE II
AVERAGE ATTACK RESULTS WITH AVERAGE ON ATTACK RESULTS ON ALL ITERATIONS (100 ITERATIONS, LEARNING RATE 10^{-3}, 10 CLIENTS).

Local Epochs	Cosine Similarty		Loss Difference		FedMIA Cosine		FedMIA Loss	
	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01
1	60.7% (+- 5.2%)	3.6% (+- 4.3%)	63.1% (+- 6.8%)	7.1% (+- 2.9%)	**72.0% (+- 4.0%)**	**10.4% (+- 5.3%)**	**69.4% (+- 4.0%)**	**10.3% (+- 3.1%)**
2	64.1%	7.1%	68.6%	14.3%	68.9%	**18.6%**	**70.5%**	12.9%
4	71.8%	12.9%	74.8%	11.4%	72.5%	**21.4%**	**76.1%**	20.0%

TABLE III
AVERAGE ATTACKS RESULTS IN THE PASSIVE SCENARIO BY CLIENT COUNT (100 ITERATIONS, LEARNING RATE 10^{-3}).

Data Distribution	N° Clients	Cosine Similarty		Loss Difference		FedMIA Cosine		FedMIA Loss	
		AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01
Overlapped	5	**66.9%**	5.7%	58.5%	5.7%	66.6%	**12.9%**	65.8%	8.6%
	10	60.7%	3.6%	63.1%	7.1%	**72.0%**	**10.4%**	69.4%	10.3%
	20	66.0%	17.1%	67.7%	17.1%	**76.8%**	**20.0%**	72.3%	14.3%

B. Proactive Attacker Scenario

1) Overall Attacks Effectiveness: The first row of Table II summarizes the results obtained with 10 different seeds to assess the reproducibility on the configuration with 10 clients, 1 local epoch, 100 iterations with a learning rate of 10^{-3}. For this configuration we provide the mean value and the standard deviation.

FedMIA-based (All VS Target) strategy clearly outperforms the 'Server vs Target' strategy for both attack signals. FedMIA Cosine and FedMIA Loss achieve an average AUC of 72% and 69.4% respectively with an average TPR@FPR of 10.4% and 10.3% respectively. In contrast, both the Cosine Similarity and Loss Difference attacks perform poorly, with an average AUC of 60.7% and 63.1% respectively with an average TPR@FPR of 3.6% and 7.1% respectively.

2) Impact of Local Epochs: Table II presents the influence of the number of local training epochs on the effectiveness of membership inference attacks. In practice, increasing the number of local epochs reduce the overall number of iteration needed, and thus reduce the communication cost of federated learning. Until we reach a number of local epoch that cause to much divergence on the local updates, preventing the convergence of the federated learning process.

Overall, we observe that attacks are sensitive to the number of local epochs. This is particularly noticeable for the 'Server vs Target' strategy, namely the Cosine Similarity and the Loss Difference attacks those AUC at 4 local epochs almost reach the FedMIA-based attacks. However, FedMIA-based attacks have a TPR@FPR that increases far above the 'Server vs Target' strategy, reaching more than 20% when the Cosine Similarity and Loss Difference remain below 13%. This result aligns with expectations: with more local updates before aggregation, the model drifts further from the global average, increasing its capacity to memorize training examples and thus making membership inference easier.

3) Influence of Client Count: Table III presents the results of our study on the influence of the number of clients in the proactive scenario. It reveals that increasing the number of shadow clients leads to much stronger attack success. As we can see, this configuration benefits FedMIA-based attacks in particular. With 20 clients, FedMIA Cosine achieves an AUC of 76.8% and a TPR@FPR of 20.0%, indicating a critical privacy breach. This trend aligns with theoretical expectations: as the number of clients grows, the statistical tests gain power, enabling the attacker to more accurately distinguish members from non-members.

Interestingly, in this overlapped setting, gradient-based attacks (FedMIA Cosine) consistently outperform loss-based methods. This setup also reflects a realistic yet concerning scenario: by adding artificial clients, a malicious server could improve its inference power by creating more "non-member" profiles to contrast against targeted clients.

4) Influence of Training Dynamics: We analyzed how model convergence and overfitting affect membership inference success by training the segmentation model with a small learning rate (10^{-4}), 10 clients, and 1 local epoch. This configuration slows down convergence, allowing us to assess attack performance throughout the learning process.

In the left of Figure 7, we display the target client model's performances after each iteration on its training dataset (named target) and the test dataset. It shows that the model starts overfitting on the target dataset at around 250 iterations. In the

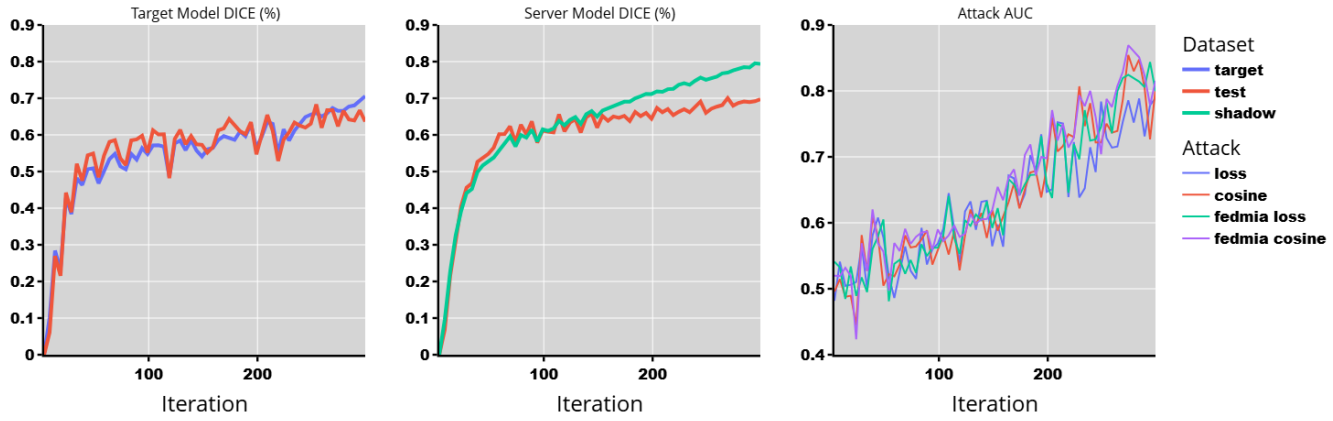


Fig. 7. Target model DICE score (left), server model DICE score (center) and attacks AUC (right) over training iterations (lr=10^{-4}). The 'dataset' legend corresponds to the plot on the left and on the center. The 'attack' legend corresponds to the plot on the right.

TABLE IV
AVERAGE ATTACKS RESULTS BY DATA DISTRIBUTION AND CLIENT COUNT (100 ITERATIONS, LEARNING RATE 10^{-3}). THE ATTACK IS PERFORMED USING AS NON-MEMBER THE TEST DATASET ONLY. THE ATTACK CONSIDER EITHER ALL ITERATIONS OR SELECTED ITERATIONS BASED ON DICE METRICS WITH 0.4 THRESHOLD.

Non member dataset	Data Distribution	N° Clients	Cosine Similarty		Loss Difference		FedMIA Cosine		FedMIA Loss	
			AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01	AUC	TPR@FPR0.01
all iterations	Normal	3	**57.9%**	**6.5%**	53.3%	1.1%	54.3%	4.3%	57.1%	5.4%
		5	55.1%	1.8%	56.4%	0.0%	58.0%	7.1%	**62.2%**	**10.7%**
	Overlapped	5	57.3%	**17.1%**	51.2%	5.7%	54.7%	10.0%	**59.0%**	7.1%
		10	56.0%	5.4%	55.6%	6.0%	**60.5%**	**9.1%**	60.2%	6.1%
		20	61.8%	5.7%	56.8%	12.9%	**66.0%**	**18.6%**	60.9%	14.3%
selected iterations	Normal	3	**62.8%**	3.2%	50.5%	1.1%	55.6%	4.3%	56.7%	**5.4%**
		5	**63.9%**	**17.9%**	52.0%	0.0%	58.8%	12.5%	63.4%	12.5%
	Overlapped	5	**65.3%**	4.3%	50.9%	5.7%	57.4%	**12.9%**	58.8%	5.7%
		10	60.9%	**10.7%**	54.4%	7.9%	**61.2%**	9.4%	**61.2%**	7.9%
		20	**68.9%**	**32.9%**	54.2%	12.9%	64.5%	14.3%	58.8%	17.1%

center of Figure 7, we show the global model's performances after the aggregation at each iteration on the shadow dataset and the test dataset. We observe also that the overfitting on the shadow dataset starts earlier at around 125 iterations. Correspondingly, the right graph of Figure 7 illustrates how all attacks benefit from model convergence. Performance remains weak while the model performances are low. When the model DICE is below 0.4, we observe that all the attacks have performances close to randomness. Then the attacks performances increase progressively, achieving around and above 0.80 in AUC after 300 iterations.

These results confirm a well-established observation: overfitting amplifies membership leakage. Even basic signals become highly predictive when the model starts memorizing training data, reinforcing the importance of carefully managing training dynamics in privacy-sensitive applications. However, if overfitting of the target model amplifies membership leakage, overfitting of the shadow clients created by the malicious server can also impact the attack. Indeed we evaluate the attack performance by its ability to distinguish between target dataset and a concatenation of test dataset and other clients dataset among whose the shadow clients. Table IV shows the attacks results when the member dataset is the training dataset of the target client, and the non-member dataset is the test dataset only. The first half of the table show the results when all iterations are taken into account. We see that the obtained results are much closer to randomness, only FedMIA Cosine achieve an AUC above 0.65 with a TPR@FPR at 18.6%. It means that in the proactive scenario, the malicious server is able to learn an attack very effective in distinguishing the target dataset from the shadow dataset but less effective to distinguish the target from the test dataset. However, in the following we highlight that by focusing on the training dynamics it increase the attacks performances on target VS test dataset.

5) Improved attack based on training dynamics: Based on the previous observation, regarding the correlation of the attack performances and the model segmentation metric performances, the server can enhance the attack by selecting the iterations based on the observed DICE. We use a threshold on the DICE obtained on the test dataset to ensure that the model is far from random. Table IV displays the attacks performances when the server selects only the iterations on which the global model achieves a DICE above or equal 0.4 on the test dataset. We see that the Cosine Similarity attack

is dramatically improved by this iteration selection **achieving above 60% of AUC in all data distribution and number of clients settings, and reaching a TPR@FPR of 32.9% with 20 clients**. However, the FedMIA-based strategy is not improved by the iteration selection.

VII. Defenses Recommendations

Numerous defense mechanisms have been proposed to mitigate membership inference attacks in federated learning. Early approaches rely on lightweight techniques such as data augmentation [32], MixUp [33], or gradient sparsification [34]. These methods aim to regularize training and reduce overfitting, thereby limiting the information leakage from model updates. However, they offer limited protection in white-box settings and can often be bypassed by adaptive attackers. Moreover, stronger defenses such as differential privacy [35], [36] inject noise into updates but typically induce a degradation of the model performances to be efficient.

To provide more robust privacy guarantees, cryptographic approaches such as Fully Homomorphic Encryption (FHE) [37], [38] and Secure Multi-Party Computation (SMPC) [39], [40] have been explored [41]. FHE allows each client to encrypt its model updates before sending them to the server, which can then perform aggregation directly in the encrypted domain. This ensures the server never has access to raw parameters. However, FHE remains computationally expensive and does not support non-linear operations natively, making it incompatible with sophisticated aggregation methods. On the other hand, SMPC distributes computation across several non-colluding servers, enabling secure training without exposing individual contributions. While more practical than FHE in certain scenarios, SMPC still incurs communication overhead and requires careful orchestration between parties. Both mechanisms have the notable advantage of preventing not only MIAs but also broader classes of privacy attacks.

In high-stakes applications such as defense, where sensitive imagery and operational data are involved, adopting such strong privacy-preserving mechanisms may become necessary despite their computational cost. Future work should focus on systematically evaluating these defenses to segmentation models training in a federated setting.

VIII. Discussion

We selected a realistic dataset closely resembling a defense scenario; however, its relatively small size, only a few hundreds of instances, posed certain limitations. Conducting our study on a small dataset constrained our ability to thoroughly investigate the effects of varying the number of clients. To ensure local learning remained meaningful, we had to introduce substantial overlap among the datasets assigned to the shadow clients (in scenarios involving a proactive attacker server). This necessity resulted in overfitting on the data utilized by these shadow models, which in turn compromised the effectiveness of FedMIA's strategy, as it relies significantly on the behavior of non-targeted client models. Expanding our research to encompass a larger dataset appears to be an exciting follow-up. Employing non-overlapping datasets for the shadow clients will likely improve the performance of FedMIA approach. Conversely, providing the target client with a more substantial amount of data may reduce the overall success rate of the attack.

Regarding the feasability, the main limitation for a membership inference attack is the access to the target dataset. To determine whether a given data point is a member of a client's training dataset, the server must have access to data highly similar to that client's data. Consequently, such attacks are more relevant in evaluation settings for assessing the privacy risks of federated learning configurations than in practical, real-world scenarios. However, in cases where the server has some prior knowledge about the target client's data distribution, it may be possible to collect sufficiently similar data to enable these attacks. Passive scenarios are particularly feasible in the absence of defense mechanisms, as they require low computational resources. Moreover they do not interfere in the FL process, they have no impact on the learned model. In contrast, proactive scenarios are feasible only for servers with significant computational resources and access to large datasets for training shadow models. Moreover, these attacks impact the learned model since the server would deviate from standard federated learning protocols by aggregating its own models.

Conclusion

We present the first analysis of membership inference attacks on federated binary segmentation models. Our results show that gradient-based attacks (Cosine Similarity and FedMIA Cosine), can effectively exploit training signals, particularly as the number of clients increases or when the server adopts a more proactive strategy by artificially introducing additional clients. Despite the limitation of a small dataset in our experimental settings, we were nonetheless able to effectively demonstrate privacy breaches.

These results underscore the critical need for robust defense mechanisms in federated learning, as, in the absence of such protections, servers are able to extract significant information about individual client datasets.

Finally, extending these attacks to multi-class semantic segmentation with large datasets and empirically assessing the practical cost of deploying defense mechanisms would offer a clearer picture of the trade-offs involved in protecting federated segmentation models.

Acknowledgments

This work has been carried out in the framework of the EDF-STORE project, funded by the European Union through the European Defence Fund (EDF), under Grant Agreement No. 101121405. The authors would like to thank Alix Lafont (Thales CortAIx Labs, France) for her valuable contributions to the illustrations presented in this work.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] Z. L. Teo, L. Jin, N. Liu, S. Li, D. Miao, X. Zhang, W. Y. Ng, T. F. Tan, D. M. Lee, K. J. Chua *et al.*, "Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture," *Cell Reports Medicine*, vol. 5, no. 2, 2024.

[3] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (Csur)*, vol. 55, no. 3, pp. 1–37, 2022.

[4] J. Rells and W. Joseph, "Federated learning for secure financial transactions," 2025.

[5] O. Stan, R. Sirdey, A. Boudguiga, R. F. Martin Zuber, and K. Hynek, "PRIVILEGE: PRIvacy and Homomorphic Encryption for Artificial IntElliGencE," in *CAID 2021 (Conference on Artificial Intelligence for Defense)*, 2021. [Online]. Available: https://cea.hal.science/cea-04487780v1

[6] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[7] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[8] L. Bai, H. Hu, Q. Ye, H. Li, L. Wang, and J. Xu, "Membership inference attacks and defenses in federated learning: A survey," *ACM Computing Surveys*, vol. 57, no. 4, pp. 1–35, 2024.

[9] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[10] G. Zhu, D. Li, H. Gu, Y. Yao, L. Fan, and Y. Han, "Fedmia: An effective membership inference attack exploiting" all for one" principle in federated learning," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 20 643–20 653.

[11] D. Godavarthi, D. Jose, S. N. Mohanty, M. Medani, M. Kallel, S. Abdullaev, and M. I. Khan, "Federated learning-based semantic segmentation framework for sustainable development," *Egyptian Informatics Journal*, vol. 30, p. 100702, 2025.

[12] Z. Zhang and G. Li, "Uav imagery real-time semantic segmentation with global–local information attention," *Sensors*, vol. 25, no. 6, p. 1786, 2025.

[13] L. Yuan, Z. Wang, L. Sun, P. S. Yu, and C. G. Brinton, "Decentralized federated learning: A survey and perspective," 2024. [Online]. Available: https://arxiv.org/abs/2306.01603

[14] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, vol. 2, no. 2, pp. 15–18, 2016.

[15] A. El Mrini, E. Cyffers, and A. Bellet, "Privacy Attacks in Decentralized Learning," in *ICML*, 2024.

[16] J. Li, N. Li, and B. Ribeiro, "Effective passive membership inference attacks in federated learning against overparameterized models," in *The Eleventh International Conference on Learning Representations*, 2023.

[17] U. Gupta, D. Stripelis, P. K. Lam, P. Thompson, J. L. Ambite, and G. Ver Steeg, "Membership inference attacks on deep regression models for neuroimaging," in *Medical Imaging with Deep Learning*. PMLR, 2021, pp. 228–251.

[18] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.

[19] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "Poisongan: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3310–3322, 2020.

[20] A. Pustozerova and R. Mayer, "Information leaks in federated learning," in *Proceedings of the network and distributed system security symposium*, vol. 10, 2020, p. 122.

[21] G. Pichler, M. Romanelli, L. R. Vega, and P. Piantanida, "Perfectly accurate membership inference by a dishonest central server in federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 4290–4296, 2023.

[22] T. Nguyen, P. Lai, K. Tran, N. Phan, and M. T. Thai, "Active membership inference attack under local differential privacy in federated learning," *arXiv preprint arXiv:2302.12685*, 2023.

[23] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1102–1107.

[24] A. Suri, P. Kanani, V. J. Marathe, and D. W. Peterson, "Subject membership inference attacks in federated learning," *arXiv preprint arXiv:2206.03317*, 2022.

[25] G. Zhu, D. Li, H. Gu, Y. Han, Y. Yao, L. Fan, and Q. Yang, "Evaluating membership inference attacks and defenses in federated learning," *arXiv e-prints*, pp. arXiv–2402, 2024.

[26] Y. Gu, Y. Bai, and S. Xu, "Cs-mia: Membership inference attack based on prediction confidence series in federated learning," *Journal of Information Security and Applications*, vol. 67, p. 103201, 2022.

[27] L. Zhang, L. Li, X. Li, B. Cai, Y. Gao, R. Dou, and L. Chen, "Efficient membership inference attacks against federated learning via bias differences," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, 2023, pp. 222–235.

[28] Y. He, S. Rahimian, B. Schiele, and M. Fritz, "Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 2020, pp. 519–535.

[29] T. Chobola, D. Usynin, and G. Kaissis, "Membership inference attacks against semantic segmentation models," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 2023, pp. 43–53.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[31] S. Waqas Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. Shahbaz Khan, F. Zhu, L. Shao, G.-S. Xia, and X. Bai, "isaid: A large-scale dataset for instance segmentation in aerial images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 28–37.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[33] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[34] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[35] Q. Zheng, S. Chen, Q. Long, and W. Su, "Federated f-differential privacy," in *International conference on artificial intelligence and statistics*. PMLR, 2021, pp. 2251–2259.

[36] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[37] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021.

[38] H. Shi, Y. Jiang, H. Yu, Y. Xu, and L. Cui, "Mvfls: multi-participant vertical federated learning based on secret sharing," *The Federate Learning*, pp. 1–9, 2022.

[39] H. Zhu, R. S. M. Goh, and W.-K. Ng, "Privacy-preserving weighted federated learning within the secret sharing framework," *IEEE Access*, vol. 8, pp. 198 275–198 284, 2020.

[40] S. S. Tiwari, G. Dhasmana, H. M. Al-Jawahry, A. Rana, G. Bhardwaj, and A. P. Srivastava, "Federated learning strategies for privacy-preserving machine learning models in cloud computing environments," in *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*. IEEE, 2024, pp. 1457–1462.

[41] O. Stan, V. Thouvenot, A. Boudguiga, K. Kapusta, M. Zuber, and R. Sirdey, "A secure federated learning: analysis of different cryptographic tools," in *SECRYPT 2022-19th International Conference on Security and Cryptography*, vol. 1, 2022, pp. 669–674.

Breaking SafetyCore: Exploring the Risks of On-Device AI Deployment

Guyomard Victor
Skyld AI
Rennes, France
victor.guyomard[at]skyld.io

Mathis Mauvisseau
Skyld AI
Rennes, France
mathis.mauvisseau[at]skyld.io

Paindavoine Marie
Skyld AI
Rennes, France
marie[at]skyld.io

***Abstract*—Due to hardware and software improvements, an increasing number of AI models are deployed on-device. This shift enhances privacy and reduces latency, but also introduces security risks distinct from traditional software. In this article, we examine these risks through the real-world case study of SafetyCore, an Android system service incorporating sensitive image content detection. We demonstrate how the on-device AI model can be extracted and manipulated to bypass detection, effectively rendering the protection ineffective. Our analysis exposes vulnerabilities of on-device AI models and provides a practical demonstration of how adversaries can exploit them.**

***Index Terms*—Reverse engineering, Model extraction, Adversarial examples**

I. Introduction

Today, an increasing number of AI[1] models are deployed on-device. This trend is driven by the growing computational power of modern hardware, especially with the adoption of specialized components like Neural Processing Unit (NPU)[1]. Moreover, advancements in AI software, such as quantization, have reduced both the size of models and the computational power required to run them, making on-device deployment more feasible and efficient. This form of deployment offers key advantages such as improved data privacy, since data is processed directly on the device, and reduced latency, as inference no longer depends on an internet connection[2]. However, while on-device AI deployment is becoming more popular, its security implications are still often misunderstood, especially when considering how AI differs from traditional software. This gap in understanding can lead to serious vulnerabilities particularly when AI is used in security applications such as content filtering or spam detection.

In this article, we explore the security risks associated with on-device AI deployment through the lens of a real-world case study: the exploitation of the SafetyCore application[3]. SafetyCore is an Android Google system service introduced in November 2024. It provides a service used by other Android applications: The classification of sensitive or problematic content, such as nudity in images. The content detection is performed using an AI based algorithm that is locally embedded on the device for privacy-preserving reasons. This means that the user data is not sent to a remote server, but is kept on the device. Currently, SafetyCore is only used by the Google Messages application for content moderation. However, other applications, such as WhatsApp[4], are expected to adopt it in the near future. The AI model used by SafetyCore takes an image as input and predicts whether it contains sensitive content or not. If such content is detected, the image is automatically blurred, and a warning message is displayed to inform the user that the image may contain unwanted characteristics, such as nudity. An example of the application behavior is shown in Figure 1.

Fig. 1. User interface of the Google Messages application with SafetyCore enabled. When nudity is detected by the AI model, the image is blurred and a warning message is displayed to the user.

SafetyCore relies on the AI model's ability to make accurate predictions based on pixel values in the image. Starting with the extraction of the embedded model, we demonstrate how adversaries can manipulate images to cause misclassifications, thereby rendering the protection mechanism ineffective. The objectives of this article are twofold:

- Explaining the specific risks of deploying AI models on-device, especially for readers without a background in AI.

[1]In this article, we use the term AI to specifically refer to deep learning neural networks.

- Providing a practical guide to extracting and exploiting these models in a real-world setting.

Each aspect of our analysis is illustrated using the SafetyCore attack case study. This attack has been performed on a Google Pixel 6, running Android 15 (build `BP1A.250305.019`) with SafetyCore (`com.google.android.safetycore`) version `1.0.757930370`.

We begin this article by examining the specific methods for reverse-engineering an AI model and discuss what makes AI models fundamentally different from traditional software. We then describe the pre-processing and conversion steps necessary to turn an extracted model into a targeted object. Finally, we explore intrinsic vulnerabilities of AI models and demonstrate how they can be exploited through AI based attacks.

II. The Reverse Engineering Challenge

This Section explores what makes AI models fundamentally different from standard code, and why traditional software protections are often insufficient to secure them from reverse-engineering.

A. What is Inside an AI Model?

The goal of an AI model is to perform a given task (prediction, generation) on data never seen before. It is defined by an architecture composed of layers, hyperparameters and learned parameters. Layers represent mathematical operations, often linear transformations such as matrix multiplications. Each layer has hyperparameters, whose values are fixed before training and remain unchanged. In contrast, the learned parameters, such as weights and biases, are updated throughout the training process so that the model can perform its task on new data. An example of a toy AI model is provided in Figure 2. The lifecycle of an AI model can be divided into two phases: training and inference. Training is the step where the learned parameters are updated, i.e. the network learns to adapt itself to new data. The inference step is when the model is used to perform some prediction on unseen data. At this stage, the learned parameters are fixed and no longer updated.

B. Why AI Models are Different from Classical Software?

Software is typically made of code that is compiled and deployed. This means that only the hardware instructions are present on the deployment target. **On the other hand, an AI model is usually stored in a file**. This file does not directly contain the hardware instructions, as traditional software does, but rather a serialized version of the algorithm. This serialized file defines the layer operations as well as the learned parameters needed to produce the model output. The specific implementation of those operations is handled by a separate inference engine. To parse and run a model, the inference engine associated with the model is required.

The operations used during inference are implemented by the inference engine. Thus, a model can only use a limited set of standardized layers. **When an AI model is run, the executed operations came from the same finite set of layers, regardless of the specific model architecture or parameters values.** The inference engine is contained in a compiled library and defines how the model is serialized. **A limited set of inference engines are often used to deploy AI models, and most of them are open-source**. This is because layer implementations are highly optimized for specific hardware, making it inefficient and unnecessary for each company developing AI models to re-implement them from scratch for all standard hardwares[5].

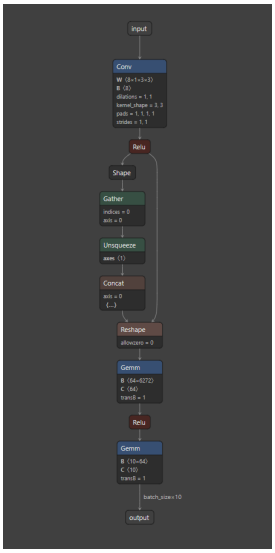


Fig. 2. Example of a toy AI model (in ONNX format). Each node of the graph corresponds to a specific layer. The first node is a convolution layer that contains learned parameters W, b and hyperparameters $dilation, kernelShape, pads, stride$.

While this standardization is beneficial for performance and cross-platform compatibility, it raises significant challenges regarding intellectual property protection.

C. Static Extraction

The first method to reverse engineer a model is to perform a static analysis (i.e., analysis without running the application) of the application to locate and extract AI models. Since serialization depends on the specific library used, it is important to understand the different libraries and their formats. The major inference engines used when deploying an AI model on-device include LiteRT[6] (formerly TensorFlow Lite), ONNX[7], and PyTorch (through TorchScript[8] and ExecuTorch[9] formats). **Each of these engines reads AI models from a file that has distinctive identifiable characteristics**. This makes it relatively easy to locate such files within a software package, enabling static file analysis. Table I lists the different characteristics that can be searched for in the files of an application to locate AI models.

a) The SafetyCore case: In the case of SafetyCore, static analysis of the service's downloaded files revealed a TensorFlow Lite model. The presence of the ASCII-encoded magic value `TFL3` at byte offset 4 led to a quick identification of the model file.

TABLE I
CHARACTERISTICS TO IDENTIFY AI MODELS USED BY MAJOR INFERENCE ENGINES.

Inference engine	Characteristic
LiteRT / TensorFlow Lite	FlatBuffers file identifier* `TFL3` (ASCII encoded)
ONNX	Protobuf containing the graph. Each layer type starts with `onnx::`
TorchScript	ZIP archive (file signature `PK\x03\x04`) containing: • The directories: – `code` – `data` • The files: – `data.pkl` – `constants.pkl`
ExecuTorch	FlatBuffers file identifier* `ET??` followed by `eh??` (ASCII encoded), where `?` is a digit.

*The FlatBuffers file identifier is a field in the FlatBuffers serialization format.

D. Dynamic Extraction, or Why Encryption Is Not Enough

In some cases, the AI model in plain-text form never touches persistent storage. For instance, when it is downloaded (remote loading) and loaded directly into memory for each inference, or stored only in encrypted form (model encryption).

- **Remote loading** avoids static interception of the model file as it is never stored in persistent storage.
- **Model encryption** effectively hides the file structure, making it impossible to locate using known characteristics of the model while performing a static analysis.

In such situations, dynamic analysis can be used to intercept the serialized model at runtime, capturing it while it is being loaded into the inference engine. This was not the case for SafetyCore, where the model was recovered through static analysis. In practice, encryption and remote loading can be bypassed, and the model extracted using dynamic analysis. While having privileged access on the running device, instrumentation tools such as Frida[10] can be used to hook the model loading functions and exfiltrate the model during execution. **Since most inference engines are open source, it is relatively easy to identify and hook the model loading function, even if it is not directly exposed by the library.**

III. AI MODEL REFINEMENT

After this first reverse-engineering step, an AI model often requires refinement before it can be effectively exploited.

A. Convert to the Right Format

Extracted AI models are often not immediately usable by attackers, because they are deployed in formats that do not support gradient computation (Additional details about gradients are provided in Section IV-A). PyTorch is the most widely used framework for attacking AI models. In contrast, formats such as TFLite and ONNX do not natively allow gradient computation, making them not suitable for direct exploitation. Therefore, converting the extracted model to PyTorch is typically a necessary step. This conversion can often be achieved using available tools, either directly or through a combination of intermediate formats. In Table II is presented common AI model formats and the corresponding tools used to convert them into PyTorch.

Original Format	Conversion Tool(s)
ONNX	`onnx2pytorch` [11]
TFLite	(`tf2onnx` + `onnx2pytorch`) via REOM[12]
TorchScript	Natively exploitable
ExecuTorch	Not currently supported*

TABLE II
CONVERSION TOOLS FOR ENABLING PYTORCH BASED ATTACKS ON EXTRACTED AI MODELS.

*Primarily due to the novelty of the framework compared to more mature alternatives such as TFLite.

B. The Quantization Problem

Quantization refers to the process of converting the learned parameters of an AI model from high-precision floating-point (typically *float32*) to lower-precision formats such as 8-bit integers (typically *int8*)[13]. This transformation reduces both memory usage and computational cost, making it particularly suitable for on-device deployment, where hardware resources are limited[13].

The most widely used approach is *affine quantization*. This method relies on two quantization parameters:

- a scale factor $s \in \mathbb{R}^+$.
- a zero point $z \in \mathbb{Z}$.

These parameters are used both to:

- Convert (quantize) the original *float32* parameters to integers.
- Compute operations directly in the quantized domain (integer domain).

Given a real-valued parameter $w \in \mathbb{R}$, its quantized representation $w_q \in \mathbb{Z}_{[\alpha_q, \beta_q]}$ is computed as:

$$w_q = \text{clip}\Big(\text{round}\big(\frac{1}{s}w + z\big), \alpha_q, \beta_q\Big)$$

Since the model no longer operates on differentiable *float32* parameters, standard gradient-based techniques cannot be directly applied to a quantized model. However, it is important to note that **quantization does not act as a security mechanism**. The combination of quantized parameters and their associated scale and zero point is sufficient to reconstruct an approximation of the original parameters. For $w \in \mathbb{R}$ we have:

$$w \approx \text{dequantize}(w_q, s, z) = s \cdot (w_q - z) \tag{1}$$

Using this equation, an attacker can construct a proxy model, i.e., a model that approximates the behavior of the original one. This proxy model is fully differentiable and can be attacked using standard gradient-based methods.

a) The SafetyCore case: In the case of the SafetyCore application, the target model was provided in the TFLite format. As shown in Table II, REOM[12] allows the conversion of a TFLite model to PyTorch by leveraging a combination of two intermediate conversion tools. Additionally, REOM integrates a quantization module that applies Equation 1 to recover *float32* parameters from *int8* parameters, enabling the construction of the proxy model. The tool successfully generated a *float32* proxy model that could be subjected to further attacks[2].

IV. Exploiting AI Models

After transforming a model into a usable artifact, we analyze the vulnerabilities of AI systems and the unique security challenges they pose. We then present how to exploit these vulnerabilities through adversarial examples. Finally, we discuss additional attacks that are relevant once an AI model is extracted.

A. Intrinsic Vulnerability of AI Models

The intrinsic vulnerability of AI relies on three intricate problems:

a) Gradient manipulation: A neural network is a highly complex function that takes an input and, using a set of parameters, produces an output. These parameters must be learned in order for the model to generate meaningful results. To learn these parameters, we define an auxiliary function, the loss function, which tells us how much the model is wrong in its prediction. For instance, in an image classification task, the loss function could quantify how inaccurately the model distinguishes between images of cats and dogs. The goal is typically to minimize this loss function. Training the model involves updating its parameters to reduce the loss, using a dataset of input/output pairs (known as the training set). This process is often performed using an algorithm called gradient descent, which iteratively adjusts the parameters in the direction that reduces the loss. The gradient is a mathematical object that indicates how to change the parameters to minimize the loss.

During the training phase, the gradients are computed with respect to the model parameters to minimize the loss. **Once the model is trained, however, an attacker can instead compute gradients with respect to the input, this time to *maximize* the loss i.e. make the model's prediction as wrong as possible. In this setting, the gradient reveals how the input should be perturbed to mislead the model.** Because neural networks are highly complex and operate in high-dimensional input spaces, these perturbations can be crafted so that they remain imperceptible to humans, making them particularly dangerous.

[2]For our proxy model, we did not add additional layers to simulate quantization errors, as we observed no significant differences between the quantized and the reconstructed model.

b) The black-box problem: Despite their remarkable performance, AI models are black-boxes in the sense that the decision-making process is not understandable by humans[14]. In other words, given a particular input, we often cannot understand why the model produces a specific output[15]. Although the field of explainable AI (XAI) has made significant progress, it remains difficult to predict how a model will behave on unseen or slightly altered inputs. This inherent opacity creates "gray areas" of unpredictable or unintuitive behavior that are exploitable. **These unpredictable behaviors are not easily identifiable or interpretable by human observers, making them ideal entry points for adversarial manipulation.**

c) Features correlation: AI models typically rely on statistical correlations in the training data rather than causal relationships. This distinction is critical: a model might learn that "A" often co-occurs with "B," without grasping whether "A" causes "B." **This reliance on correlation rather than causation contributes to unexpected and or unintelligible model behavior that can bypass human judgment.**

B. Exploiting These Vulnerabilities

The vulnerabilities presented above can be exploited in multiple ways across the AI lifecycle. In this Section, we focus on concrete attack strategies that target on-device AI models when having access to the architecture and parameters.

a) Inferring the loss function: Many attacks rely on gradient computations, which not only require knowledge of the model's architecture and parameters but also defining a suitable loss function. The choice of this loss function is driven by the identification of the task the model is solving, for example a binary classification task. **A common attack strategy is to identify the loss that was used for model training, and use the opposite for attacking (in order to maximize it). While this information is typically unavailable after deployment, attackers can often infer it through careful analysis of the architecture and model behavior.**

This process typically involves the following steps:

1) **Architecture probing**: By analyzing the metadata (e.g., input/output shapes, presence of specific layers such as convolutions or residual blocks, and even embedded strings in the model file), one can make educated guesses about the model architecture and its intended task. For some model formats, such as TFLite or ONNX, the overall architecture can be visualized using a visualization tool like Netron[3].
2) **I/O probing**: By feeding the model with various sample inputs and observing the output responses, one can understand the appropriate input format and the semantics of the outputs (classification scores, images, heatmaps etc...).
3) **Output layer inspection**: The final activation function often reveals the nature of the learning problem. A softmax activation suggests a classification task, and a

[3]http://netron.app

sigmoid (logistic) activation a multi-label classification task. A linear output usually indicates regression.

b) The SafetyCore case: In the case of the SafetyCore, the input tensor shape is $1 \times 224 \times 224 \times 3$, which strongly suggest image data. The architecture includes residual connections common in ResNet architectures[16] for image classification. The output shape is 1×4, and when probing the model with explicit versus non-explicit images, we observe that explicit content causes some output values to exceed 0.5, while benign content remains below this threshold. This, along with the presence of a sigmoid activation function before the output, suggests that the model is solving a multi-label classification problem, and was trained using a binary cross-entropy loss. This inferred loss enables the attacker to compute gradients for further manipulations.

1) Adversarial Examples: **The philosophy behind adversarial examples involves defining a desired criterion on the model's output through a loss function, and then using model gradients to find an input modification that satisfy this criterion.** This attack relies mostly on the properties of high dimensional spaces and the nature of the functions learned by deep learning models. In these spaces, a tiny perturbation, when applied in a specific direction (using gradient), can often cause a significant change in the model's output.

Generally, the input change is sought to be imperceptible and to maximize the model's output change[17]. For classifiers this could mean altering a picture of a panda so that the model confidently predict a gibbon[17]. There are two main types of adversarial attacks:

- **Untargeted attacks:** The goal is to mislead the model, regardless of what that incorrect output is.
- **Targeted attacks:** The goal is to produce a predetermined output. The attacker doesn't just want the model to be wrong, he wants it to produce a particular output.

For generating these examples, a diverse range of attack algorithms has been developed, from the fast gradient sign method (FGSM)[17] to more iterative and powerful methods like Projected Gradient Descent (PGD)[18], each with different trade-offs in terms of computational cost and effectiveness[19].

It is important to note that the effectiveness of adversarial examples depends on the data modality of the input. They are particularly effective on continuous data, such as images and audio, where small perturbations can be applied directly to the input using gradient-based methods[18]. In contrast, generating adversarial examples for discrete data (like text) is more challenging, as it is difficult to generate discrete changes in a meaningful way using gradients[20]. However, many text models include both discrete and continuous components. In such cases, it is often possible to generate adversarial perturbations in the continuous space and then extrapolate them back to the discrete space[20].

Adversarial examples are extremely powerful in practice. Because of the high dimensionality of the input space, it is not feasible to simply "patch" a given adversarial example by specifically instructing the model to ignore it during training. Doing so leaves the model vulnerable to countless regenerated variants that came from the same region of the input space. A common defense strategy is adversarial training[21], which involves incorporating multiple adversarial examples in the model training. While it offers a potential defense, it remains difficult to fully mitigate the threat, as this approach often involves a trade-off with model performance[21].

a) The Safetycore case: For SafetyCore, we implemented a **Projected Gradient Descent (PGD)** attack, a widely used method known for its effectiveness in white-box setting (e.g when access to the model parameters).

This allows two types of attacks:

1) **False Positive (Enable Blurring):** You can start from a non-explicit image and generate a small, imperceptible perturbations to make the model predicting it as explicit. As a result, SafetyCore will apply blurring to an image that should not be blurred.
2) **False Negative (Bypass Blurring)** Yo can also start from an explicit image. This image will contains a small perturbation that prevent the model from recognizing it as explicit. Consequently, SafetyCore will not apply any blurring to it, effectively bypassing the protection.

In Listing 1 a PyTorch implementation of the attack script is provided. This script is intended to be simple and as generic as possible and can be reused on other models as long as the input data is continuous, and a loss function can be chosen. The most important parameters of this script are ϵ and num_iter.

- ϵ control the maximum change per pixel that is expected for the perturbed input. Higher values mean higher pixels variations and then more visible perturbations. You can gradually increase ϵ until you find a sample with the expected model output.
- num_iter control the number of iteration (number of gradient steps) that are taken during the attack. A higher number of iterations allows finding a more effective adversarial example in terms of loss function maximization.

Before running the attack script, the input images are resized to $1 \times 224 \times 224 \times 3$ to match the model's input dimensions. The resulting adversarial examples have the same size and are saved in a .png format. It is important to use an image format that does not apply compression in order to preserve the adversarial perturbation (avoiding JPEG, which would remove part of the added noise).

In Figure 3, we show three benign images that have been perturbed using PGD (Enable Blurring case). When passed through the originally extracted model, these adversarial images are misclassified as explicit content. In Figure 3, we also present a screenshot from the Google Messages app showing how these images appear to users: all are blurred with a warning about potential explicit content. You can imagine the same scenario with explicit images that will not be blurred in the application (Bypass Blurring). For ethical reasons, this case is not illustrated in the article. Executing an "enable

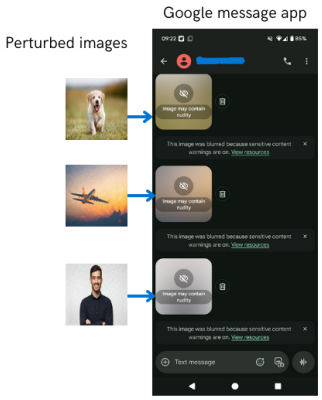


Fig. 3. Adversarial attack on SafetyCore. On the left, are provided the perturbed images obtained after a PGD attack, and on the right what appears in the Google Message application when sent to an Android device.

blurring attack" or a "bypass blurring attack" follows the same methodology, allowing the adversary to choose either at will.

The implications of this attack are severe: since the same AI model is shared across all Android devices, it is possible to fully bypass the filtering capabilities, regardless of the input. This attacks takes less than 30 lines of code.

2) Additional Relevant Attacks: Even if it is very powerful, adversarial examples generation is not the only relevant attack that can be performed after extracting an AI model. In this Section we presented two additional types of methods that can be applied.

a) Model inversion: Model inversion attacks aim to reconstruct representations of the training data by using the AI model itself[22]. Typically, these attacks exploit model gradients to iteratively optimize an input that maximizes the model's confidence[22]. This technique is most commonly applied to classification tasks, where the goal is to generate inputs that are representative of a target class. The consequences can be severe, often resulting in the leakage of private data for example, reconstructing faces that were used to train a facial recognition model. Additionally, model inversion can provide insights into the task the model was trained on. For instance, if the reconstructed inputs resemble airplanes and the model architecture indicates a classification task, one can reasonably infer that the model is likely classifying different types of planes from images.

In the context of SafetyCore, we attempted a model inversion attack. However, the optimization process quickly collapsed. This failure can be attributed to two main factors. First, the training data for each class is probably highly diverse, making it difficult for the model to converge toward a shared representation. Second, the task is a multi-class classification problem, which further complicates the inversion process, as each class may share overlapping features with others.

b) Backdoor Attacks: Backdoor attacks involve poisoning the training data with specially crafted samples that cause the model to learn spurious correlations between a trigger pattern and a specific output[23]. When the attacker later inputs a sample containing the same trigger, the model exhibits the intended behavior[23].

This type of attack exploits two vulnerabilities described in Section IV-A:

- The model is learning arbitrary associations (like small patch for images) that have no causal relation to the task.
- The black-box nature of AI means that such malicious correlations are difficult to interpret or detect after training.

In the case of SafetyCore, this type of attack may be unnecessary, as adversarial examples can achieve the same effect without requiring the model to be retrained. Moreover, there is no indication that the submitted images will be used for future training. In fact, since the model runs entirely on-device, retraining seems unlikely. However, unlike adversarial examples, which may not transfer to new model versions, backdoor examples are more likely to persist. Indeed, as backdoors are difficult to detect in training data, such examples could remain in future training sets, potentially preserving the backdoor across versions.

```python
import torch
from torch import nn

def pgd_attack(
    model, inputs, targets, epsilon=0.01, alpha
    =0.005, num_iter=100,
    loss_fn=None, random_start=True, clip_min=0.0,
    clip_max=1.0):
    """
    Projected Gradient Descent (PGD) attack on a
    PyTorch model.

    Parameters:
    -----------
    model : torch.nn.Module
        The neural network to attack.
    inputs : torch.Tensor
        Original input images or continuous data to
    perturb.
        TODO: Replace by your own input
    targets : torch.Tensor
        Ground truth labels corresponding to the
    inputs.
    epsilon : float
        Maximum perturbation allowed (L-infinity
    norm).
    alpha : float
        Step size for each iteration.
    num_iter : int
        Number of attack iterations.
    loss_fn : callable, optional
        Loss function to maximize (defaults to
    BCELoss if None).
        TODO: Replace by your own loss function
    random_start : bool
        If True, start from a random point within
    the epsilon-ball around the input.
    clip_min : float
        Minimum allowed value for perturbed inputs.
    clip_max : float
        Maximum allowed value for perturbed inputs.

    Returns:
    --------
    torch.Tensor
```

```
        Adversarially perturbed inputs.
    """
    model.eval()
    original_inputs = inputs.clone().detach()

    if random_start:
        # Start from a random point within the epsilon-ball
        adv_inputs = original_inputs + torch.empty_like(inputs).uniform_(-epsilon, epsilon)
        adv_inputs = torch.clamp(adv_inputs, clip_min, clip_max)
    else:
        adv_inputs = original_inputs.clone().detach()

    if loss_fn is None:
        # Loss function used for the attack
        # TODO: Replace by your own loss function
        loss_fn = nn.BCELoss()

    for _ in range(num_iter):
        adv_inputs.requires_grad_(True)
        outputs = model(adv_inputs)
        loss = loss_fn(outputs, targets)

        model.zero_grad()
        loss.backward()
        grad_sign = adv_inputs.grad.detach().sign()

        adv_inputs = adv_inputs + alpha * grad_sign
        # Project back to the epsilon-ball and clip to valid range
        perturbation = torch.clamp(adv_inputs - original_inputs, min=-epsilon, max=epsilon)
        adv_inputs = torch.clamp(original_inputs + perturbation, clip_min, clip_max).detach()

    return adv_inputs
```

Listing 1. Small generic Python code for generating adversarial examples with PGD

V. Conclusion

Security should serve as a cornerstone for building trust in AI systems. In this paper, we explored the risks of deploying AI models on-device through the lens of the SafetyCore application. Our work demonstrates that once adversaries gain access to the model, it can be compromised with relative ease and rendered ineffective, raising important concerns for the security of on-device AI based applications. While on-device AI enables countless use cases, its specific security challenges are still overlooked, even by major players in the field as illustrated by the SafetyCore example. This work acts as a foundation for understanding and running attacks on on-device AI models, and can be extended to a wide range of applications. In future work, we plan to extend our methodology to more data modalities and use-cases.

References

[1] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12): 2295–2329, 2017. doi:10.1109/JPROC.2017.2761740.

[2] Kimberly Jane. Edge ai devices for multimodal document intelligence: Designing low-latency, privacy-preserving systems for on-device fraud prevention. 03 2025.

[3] Google. Safetycore (android system), 2025. URL https://play.google.com/store/apps/details?id=com.google.android.safetycore&hl=fr.

[4] Stan Kaminsky. Whatsapp integration, 2025. URL https://www.kaspersky.fr/blog/what-are-android-safetycore-and-key-verifier/22653/.

[5] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel Emer. Efficient processing of deep neural networks: A tutorial and survey, 2017. URL https://arxiv.org/abs/1703.09039.

[6] Google. Litert inference engine, 2025. URL https://ai.google.dev/edge/litert?hl=fr.

[7] Microsoft. Onnx inference engine, 2025. URL https://onnx.ai/.

[8] Meta. Torchscript inference engine, 2025. URL https://docs.pytorch.org/docs/stable/jit.html.

[9] Meta. Executorch inference engine, 2025. URL https://docs.pytorch.org/executorch/stable/index.html.

[10] Frida. https://frida.re/. Dynamic instrumentation toolkit.

[11] ENOT developers, Igor Kalgin, Arseny Yanchenko, Pyoter Ivanov, and Alexander Goncharenko. onnx2torch. https://enot.ai/, 2021. Version: x.y.z.

[12] Mingyi Zhou, Xiang Gao, Jing Wu, Kui Liu, Hailong Sun, and Li Li. Investigating white-box attacks for on-device models, 2024. URL https://arxiv.org/abs/2402.05493.

[13] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization, 2021. URL https://arxiv.org/abs/2106.08295.

[14] Zachary C. Lipton. The mythos of model interpretability, 2017. URL https://arxiv.org/abs/1606.03490.

[15] Victor Guyomard, Françoise Fessant, Tassadit Bouadi, and Thomas Guyet. Post-hoc counterfactual generation with supervised autoencoder. pages 105–114, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi:10.1109/CVPR.2016.90.

[17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. URL https://arxiv.org/abs/1706.06083.

[19] Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*, 12:61113–61136, 2024. ISSN 2169-3536. doi:10.1109/access.2024.3395118. URL http://dx.doi.org/10.1109/ACCESS.2024.3395118.

[20] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification, 2018. URL https://arxiv.org/abs/1712.

06751.

[21] Weimin Zhao, Sanaa Alwidian, and Qusay H. Mahmoud. Adversarial training methods for deep learning: A systematic review. *Algorithms*, 15(8), 2022. ISSN 1999-4893. doi:10.3390/a15080283. URL https://www.mdpi.com/1999-4893/15/8/283.

[22] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450338325. doi:10.1145/2810103.2813677. URL https://doi.org/10.1145/2810103.2813677.

[23] Yang Bai, Gaojie Xing, Hongyan Wu, Zhihong Rao, Chuan Ma, Shiping Wang, Xiaolei Liu, Yimin Zhou, Jiajia Tang, Kaijun Huang, and Jiale Kang. Backdoor attack and defense on deep learning: A survey. *IEEE Transactions on Computational Social Systems*, 12(1): 404–434, 2025. doi:10.1109/TCSS.2024.3482723.

Red Team BERT: Reliable Automated Penetration Testing with Multi-BERT Architecture

Christophe Genevey-Metat
R&D AI / Cyber Team
NEVERHACK
Rennes, France
cgeneveymetat@neverhack.com
0000-0002-1901-626X

Guillaume Nollet
R&D AI / Cyber Team
NEVERHACK
Rennes, France
gnollet@neverhack.com
0009-0006-7101-8680

Pierre-Marie Satre
R&D AI / Cyber Team
NEVERHACK
Rennes, France
pmsatre@neverhack.com
0000-0002-8010-1885

Loïc Scotto Di Perrotolo
R&D AI / Cyber Team
NEVERHACK
Rennes, France
lscottodiperrotolo@neverhack.com
0009-0001-8614-7008

Olivier Gesny
R&D AI / Cyber Team
NEVERHACK
Rennes, France
ogesny@neverhack.com
0000-0003-2132-4875

***Abstract*—Artificial intelligence has become particularly crucial in its integration with cybersecurity applications. Numerous studies have demonstrated that reinforcement learning agents can identify optimal sequences of actions to attack a network. However, these agents are often overfitted to specific environments and struggle to generalize or adapt to networks that differ from those encountered during training. Moreover, most of these agents rely on complex abstract architectures which lack interpretability in their decision-making processes. In this work, we propose a novel agent architecture that integrates four transformer-based BERT models. This architecture can adapt to any given network, is robust to changes in parameters and objectives, and requires no additional training phase to attack unseen networks. Additionally, our model is interpretable using SHAP values. We introduce a specific context tailored to this architecture and reuse an evaluation metric we developed in our previous work that more accurately reflects an agent's ability to attack a network without prior knowledge. Finally, we discuss the limitations of our approach and outline future directions to facilitate its deployment in real-world applications.**

***Index Terms*—pentesting automation, attack simulation, zero-shot classification, transformers, large language model, adaptability, robustness, explainability**

I. Introduction

Pentesting consists of discovering and exploiting weaknesses within a network, in order to help a company identify vulnerabilities present in their information system. In the context of research, pentesting is always studied as an optimization problem whose objective is to compromise a target with a minimum sequence of actions. Most agents are trained using reinforcement learning, and they choose unit attacks to reach and compromise the target. Many researchers [1], [4]–[7], [9]–[11] have studied RL agents to solve pentesting problems, and several simulators have been developped to help the community train their own RL agents. However, in a real-world application, pentesting is usually more complex than an optimization problem, because network topologies differ across companies: a sequence of actions that works for one organization may fail for another due to specific configurations. Thus an AI agent must explore and discover information before exploiting it. This discovery of information results in extra actions being performed instead of blindly learning the path towards targets.

In our previous work [2], we presented a zero-shot LLM-based agent for automated network penetration testing, offering strong adaptability without retraining. We introduced a novel evaluation metric that rewards informed decision-making over trial-and-error. The model outperformed classical RL agents (DQN, PPO, CLAP) in unseen NASim [8] environments of small and moderate sizes using textual observations. Preliminary SHAP-based analysis provided insight into the agent's decision-making process. Overall, this served as a promising step towards explainable and robust AI for red teaming.

In this paper, we extend our approach by combining several BERT models to build a complete pipeline that determines, at each point in time, the machine one must target and the action that needs to be taken. First, we construct various datasets derived from the NASim environment, with dynamically generated network topologies. We train our BERT models separately: one to predict the correct machine given a textual representation of the network; one to predict the appropriate action based on the selected machine; and two to predict eventual parameters of the action. Each textual representation is distinct and tailored to its respective decision-making task. After training the four BERT models, we develop a sequential pipeline to execute them in inference mode. This pipeline is then used to predict a full sequence of actions on unseen NASim topologies. To improve generalization, we also implement a rollout mechanism and a padding system, allowing the pipeline to adapt to new topologies that may be smaller or larger than those encountered during training.

Finally, we compare our approach to another reinforcement learning-based method called CLAP [10], and our previous work on zero-shot decision making. We demonstrate that our pipeline outperforms these two RL agents and shows much stronger generalization capabilities. We also provide explainability results on our BERT-based architecture and compare them with our initial explainability work on the zero-shot model.

The paper is organized as follows: Section II presents related works on RL agents for pentesting. Section III introduces the environment and data used in our approach. Section IV presents our four BERT architectures for machine, action, and parameter selections, and the specific contexts of each of them. Section V presents our experiment on the explainability obtained after training. Section VI details the performance and results of our pipeline in an RL environment and compares them to other RL agents. Finally, Section VII concludes with future improvements and works to extend our BERT-based system.

II. Related works

The research community has focused on two main areas: generating environments for scenario simulation and emulation, and designing models for efficient real-life deployment.

Famous environments for automating red team exercises include NASim [8] and NASimEmu [5]. NASim (literally "Network Attack Simulator") simulates an information system that is composed of one or multiple subnetworks, each containing a specific number of host machines. Each host has an operating system and exposes services and processes that may be scanned and targeted to gain privileged access to the host. The main goal of NASim scenarios is to gain full access on specific hosts, which are called targets. Firewalls impose restrictions on both inter-subnet traffic and Internet connectivity. NASimEmu builds on the NASim environment to support emulation, and may also generate scenarios with a given level of diversity by changing the path required to reach the target. Other works for environment simulation include the GAP model [11] introduced by Zhou et al. in 2025, which explores domain randomization for scalable scenario generation and meta-reinforcement learning to enable few-shot adaptation. A Large Language Model is used to perform domain randomization and generate diverse environments that are closer to real-world network configurations.

Most of the literature adapts various reinforcement learning methods to the problem of automated pentesting, but most of them show weaknesses regarding generalization difficulties. For instance, in 2022, Yan et al. presented CLAP [10], a variant of the PPO algorithm that can handle multi-objective reinforcement learning in a pentesting context. CLAP uses a cover mask mechanism that allows the model to keep track of previous actions performed in the past. Though the authors show that their model quickly converges to the optimal sequence over the three scenarios during the training phase compared to other algorithms (DQN, Improved-DQN, HA-DQN), they do not show the performance during an evaluation phase. The same goes for AutoRed [3], presented in 2024 by Hasegawa et al., which utilizes Graph Neural Networks (GNNs) to encode the dynamic structure of the target network. This approach was not evaluated on previously unseen topologies, making it unclear whether the agent can generalize effectively to new network structures.

To tackle this challenge of generalization, Yan et al. introduced SetTron [9], which uses randomly-rearranged state representations that include information about each host as well as actions performed on them. SetTron is more flexible than CLAP in that it maintains good performance if one changes the location of the target in the network on which SetTron has been trained. However, the adaptability of the model for truly unseen network topologies has not been demonstrated by the authors. Another approach [7], presented by Nyber et al. in 2024, goes a bit further: a message-passing neural network (MPNN) is trained with multiple agents using reinforcement learning. Evaluated in the CAGE 2 environment, it achieves non-trivial performance on scenarios which it has not been trained on, although this performance remains inferior to Multi-Layered Perceptrons (MLP) trained specifically on these scenarios.

Finally, a certain amount of papers defend the use of Large Language Models (LLMs). On the defensive side, Huan et al. presented in 2024 a two-step LLM called PenHeal [4], that first identifies multiple vulnerabilities in a system, and then suggests optimal remediation strategies. In 2025, Kim et al. introduced a model called CyberAlly [6], which aims to provide real-time, context-aware support for cybersecurity professionnals through textual clues. On the more offensive side, in 2024, Deng et al. introduced PentestGPT [1], a multi-agent LLM planner that automates three key red team processes: planning, execution, and result interpretation, using one LLM for each. The three modules work together to guide the user during the penetration testing process. Finally, in our previous work [2], we introduced a zero-shot LLM-based approach for automated pentesting, which outperformed state-of-the-art RL methods like DQN, PPO, and CLAP on NASim benchmarks and also required no retraining to generalize across unseen topologies. A new evaluation method rewards the discovery of useful information for exploits and privilege escalations, thus promoting better decision-making. Early explainability results using SHAP highlight its potential for interpretable red teaming.

III. Original data, featuring and actions

A. General concept

Selecting an attack target requires enumerating available machines with their properties and predicting the most urgent one. The same principle applies to action selection: given the properties of all available machines and the list of available actions, the most critical action must be predicted. Note that a third choice may be required for actions that require specific parameters (such as exploiting a particular service or escalating privileges on a specific process). In that case, the context of the machine, the chosen action, and the list of

available parameters is given for that final choice. These three choices lead to the definition and training of three separate BERT models.

NASim is ideal for handling some real-world complexities (e.g. firewall restrictions, non-deterministic actions), even though its overall complexity remains much lower than that of an actual real-world application. In the case of NASim, the information required to determine which machine should be attacked encompasses the IP address, role, current access, observed OS, observed processes and observed services of every machine. It is easy to extract this information from the NASim environment and print it sequentially.

To be more precise, the NASim environment stores at each point in time a list of machines, each identified by IP address and characterized by the following: operating system, available services and processes, and status indicators for discovery, reachability, sensitivity classification, and our current level of compromise. We have augmented it with indicators showing whether subnet scans have been launched from that machine.

Once a machine has been selected, the appropriate action to perform on the target machine must also be selected from the various action types that NASim supports:

- Simple actions to collect more information: OS scan, process scan, service scan, subnet scan
- Complex actions aimed at gaining restricted access on a machine: vulnerability exploit, privilege escalation

Complex actions need parameters in order to specify what type of attack we want to execute. For vulnerability exploit-type attacks, these parameters consist of a required service, an optionally required OS, as well as the access granted by the attack. For privilege escalation type attacks, the parameters consist of a process, the given access, and optionally an operating system.

B. Abstraction methods

Pentesters are primarily concerned with actionable vulnerabilities rather than exhaustive system analysis, which implies cross-referencing the vulnerabilities that stem from the processes, services and operation systems of the machine. For this reason, our transformer focuses on linking relevant concepts rather than understanding domain-specific details. For instance, let's imagine an exploit that uses the FTP service in a Linux OS to grant the User access to the attacker. Though this representation appears meaningful, a model only needs to understand that an exploit that uses a service called X, an OS called Y to grant access Z is relevant to be used if the target machine uses service X on OS Y, and access Z has not yet been granted on it.

For this reason, and in order to get results that may be more generalizable and less domain-specific, a conversion is applied to most domain-specific words in the source text at random. "Linux" and "Windows" may be converted to "OS1" and "OS2", respectively, or the other way around. Similarly, services "FTP", "SSH" and "HTTP" may become any of "SERVICE1", "SERVICE2" and "SERVICE3" in any order. The key requirement is that within any given sentence, each service is represented consistently and distinctly from other services.

Sentence	Meaning
IP2 STATUS1 ROLE1 OS1 SERVICE2 SERVICE3 PROCESS0	The machine with IP n°2 is not critical (ROLE1) and has not been attacked yet (STATUS1). It uses operating system 1, houses services 2 and 3, and its processes are not yet known. Machine selection consists of many such sentences separated by commas.
ROLE1 ACCESS0 OS0 SERVICE2 PROCESS0 | PRIVILEGE, OSSCAN, EXPLOIT, DISCOVERY, PROCESSSCAN	The machine we target is not critical (ROLE1), does not have any privileged access (ACCESS0), and we don't know its operating system or processes. We do know that it houses service 2. The five actions listed after the pipe may be launched. This sentence is used for action selection.
ACCESS1 OS1 SERVICE5 SERVICE2 PROCESS2 | PRIVILEGE | PARAM4 PROCESS3 ACCESS2, PARAM1 PROCESS2 ACCESS2	Considering the machine with given characteristics and the fact that a privilege escalation action is about to be launched, two parameters are available: an escalation using process 3, another using process 2. Both grant access 2. This sentence is used for parameter selection.

TABLE I
INPUT SENTENCE EXAMPLES FOR MACHINE, ACTION AND PARAMETER SELECTION.

Table I describes the three types of sentences or propositions that are used in our classification processes.

Of the three classification problems, the one whose inputs tend to be the longest is the machine classification problem. Indeed, if one needs to differentiate between four different machines, one needs to describe all four machines fully. Note that, with this kind of convention, it is possible to identify a machine using either its position in the list (select the fourth machine in the list), or its identifier (select the machine with IP16 - we use the term "IP" loosely here to refer to the machine identifier). In the following section, when defining the transformer architecture for machine selection, we compare the results of our architecture in both cases and point out the advantages and drawbacks of each of them. The same is done for parameter selection.

C. Dataset generation

In our study, we generated both synthetic datasets (generated using hard-coded rules) and simulated datasets (generated by running NASim on different networks and saving the observed environment at each step). The results shown in sections IV and VI only use simulated datasets, and no synthetic dataset has been used to train nor test the models we present in this paper.

In order to generate a labeled dataset for the different parts, we design a heuristic that assigns the labels corresponding to the 'best machine', the 'best action', and the 'best parameter'.

We define the "best machine" to select using the following rules:

- first select any machine that is among the target machines of the network,

- then select any machine which has been compromised but not fully exploited (a machine is considered fully exploited once a subnet scan has been executed from it. For target machines in the network, root access needs to have been obtained as well),
- then select any machine which has not yet been compromised.

Once a machine has been selected, we define the "best action" using the following rules:

- scan the machine OS and services if the machine is not yet compromised,
- launch an exploit if all the information needed to launch one is discovered,
- launch a process scan if the machine is a target and is compromised,
- launch a privilege escalation if the machine is an already compromised target machine and processes have been scanned,
- launch a subnet discovery if the machine is fully compromised (for targets) or compromised in any way (for non-targets).

The selection of parameters ensures consistency by choosing exploits that match the target's operating system and use the services available on the machine. When multiple compatible exploits are available, the selection prioritizes the one that grants the highest privilege level. A similar reasoning is used for privilege escalation actions.

We execute this heuristic on random NASim scenarios that have a number of hosts equal to 4, and a number of services equal to 4.

In Table II, we summarize the number of challenges executed using the heuristic in order to generate the datasets. We also summarize the amount of labels inside each dataset.

TABLE II
NUMBER OF CHALLENGES USED TO MAKE DATASETS

Dataset For	Train	Valid	Test	Labels
Machine selection	100K	50K	5K	10
Actions selection	100K	50K	5K	6
Param selection (exploit)	50K	50K	5K	4
Param selection (privilege)	100K	50K	5K	2

IV. TRANSFORMER DEFINITION AND RESULTS

Since all three problems (machine selection, action selection, and parameter selection) rely on textual input representations, transformer architectures are particularly well-suited to solve them. We train four distinct transformer models for this purpose:

- one for machine selection
- one for action selection
- one for parameter selection in the case of an exploit
- one for parameter selection in the case of a privilege escalation

Each transformer is trained using the labeled data from the aforementioned databases. During training, we constrain our dataset to randomly generated NASim networks with 4 hosts. Generalization to larger networks is explored in section VI. This training proves highly effective, with every transformer achieving 100.0% accuracy on both validation and test datasets.

Table I details the inputs for each transformer. The final entry regarding exploit and privilege escalation parameters applies to both parameter selection transformers: one processes examples containing "EXPLOIT" keywords, while the other handles examples with "PRIVILEGE" keywords.

TABLE III
HYPERPARAMETER COMPARISON BETWEEN THE FOUR BERT MODELS AND THE ZERO-SHOT ARCHITECTURE.

Parameter	BERT	Zero-shot
Epochs	[40, 10, 40, 40]	None
Batch Size	[32, 32, 32, 32]	None
Vocab size	[37, 45, 29, 35]	50,265
Parameters	[486K, 486K, 495K, 495K] = 1.97M	407M

In Table III, we summarize some of the hyperparameters of our four BERT models (machine selection, action selection, "EXPLOIT" parameter selection and "PRIVILEGE" parameter selection respectively), and compare them to our previous work based on a zero-shot model [2]. When comparing the two approaches, we observe that our four BERT models combined have a much smaller amount of parameters than that of the zero-shot model, resulting in a more lightweight architecture.

In the next section, we demonstrate that our model provides more consistent interpretability compared to the zero-shot model.

V. EXPLAINABILITY OF THE RESULTS

Hereafter, we present two methods for model explainability. The first is SHAP values, a game-theoretic approach to explain the influence of certain features using do-calculus. The second approach involves systematically modifying inputs in domain-relevant ways to observe our model's responses.

SHAP values quantify each input feature's contribution to the prediction using game theory. The SHAP value of a model relative to a given input feature corresponds to the expected value of the output of that model as we intervene on the input feature, and represents how much that feature influences the model's output compared to the baseline prediction. There is a set of SHAP values for each predicted class : one may both analyze the reasons why a certain output has been chosen, and why another has been rejected.

A. Machine selection

In this section, we demonstrate how role and status keywords influence our model's decision-making. To illustrate this, we created two different network contexts with four potential IPs to choose from.

In the first context, three machines share the same information (ROLE1 and STATUS1), while one has a different status

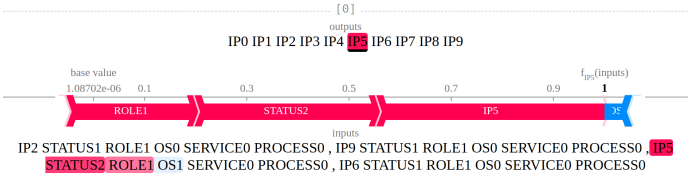


Fig. 1. Shap value of label "IP5"

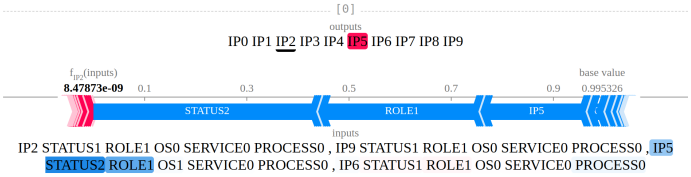


Fig. 2. Shap value of label "IP2"

(STATUS2), indicating that it is already under attack. That machine is correctly predicted (IP5) by the model. In each figure in this section, one can observe possible outputs at the top of the picture and the model input at the bottom. For a selected output (which is underlined), we can observe the contributions of each input token. Figure 1 shows that the words IP5, STATUS2 and ROLE1 from the target machine provide the strongest positive contributions to this prediction, which is consistent with human reasoning.

When examining the other predicted labels, we see little to no influence overall, except for label IP2 (Figure 2), where STATUS2 and ROLE1 from the third machine (IP5) contribute negatively. This negative contribution is also meaningful, as it reduces the confidence in IP2 when machines with higher-priority statuses are present in the input.

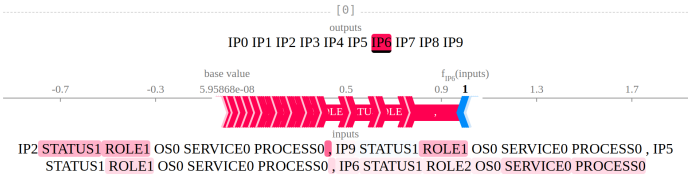


Fig. 3. Shap value of label "IP6"

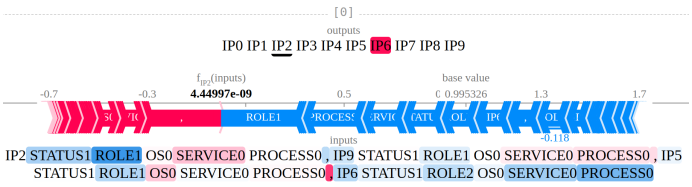


Fig. 4. Shap value of label "IP2"

The second context is quite similar: three machines share the same attributes (ROLE1 and STATUS1), and one machine also has STATUS1 (indicating no attack) but is associated with ROLE2 (indicating a potential target). The predicted label is IP6, which corresponds to the target machine. Interestingly, the SHAP values from Figure 3 indicate that the model's positive attribution is not primarily based on ROLE2, but instead on the STATUS1 and ROLE1 from the other machines in the input. This result can be interpreted as follows : if all others machines have ROLE1 but IP6 has ROLE2, then it is IP6 that must be selected. Further insights are provided in Figure 4, which presents the SHAP values for the alternative prediction IP2. We observe several negative contributions, notably from ROLE1 and STATUS1, which appropriately reduce the model's confidence in selecting IP2. This behavior is logical: when a more relevant role (ROLE2) is present elsewhere in the input, candidates with less relevant roles should receive lower confidence scores. Additionally, ROLE2 associated with the IP6 machine also contributes negatively to the prediction of IP2, reinforcing the model's preference for IP6.

We note, however, that in both Figure 3 and Figure 4, words other than role and status turn out to be important during the classification process. In particular, we observe in this context that the word ',' is one of the main positive factors in the prediction of the IP. This specific case shows us that our model's learning is not perfect, and that we must remain cautious regarding its interpretation. We verified these observations using direct intervention on inputs generated using live scenarios and looking at the response. In doing so, we noticed that downgrading the role and status of the selected machine to role 1 (unimportant) and status 1 (unattacked), when there are other machines with a better role-status combination, changes the prediction of the model 69.4% of the time. Had the model only used role and status to make its decision, we would have expected a 100% score instead. The difference lies in the fact that other parts of the input data are used, most probably because of the important correlation that happens in practice between the status and the information we have access to (OS, services and processes).

B. Action selection

This section demonstrates how operating system and services features influence the action selection process. For this demonstration, we constructed a context with a maximum of six selectable actions.

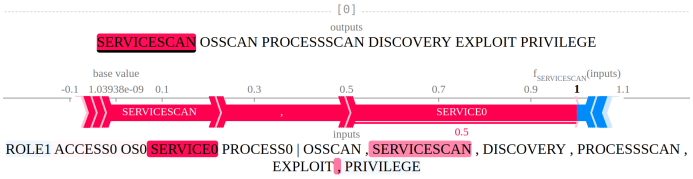


Fig. 5. Shap value of label "SERVICESCAN"

In this scenario, our model does not have any type of information regarding the host: OS0, SERVICE0 and PROCESS0 indicate that the OS, services and processes are unknown. For this reason, the predicted action is a SERVICESCAN. In Figures 5 and 6, we observe that the most significant

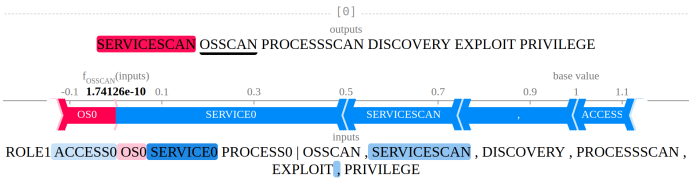


Fig. 6. Shap value of label ”OSSCAN”

contributions come from SERVICE0 and the action name SERVICESCAN itself. Since it is both feasible and necessary to execute a service scan, the model appropriately selects SERVICESCAN over other available actions. Note that, in Figure 6, the OS0 word contributes positively to the OSSCAN prediction, which makes sense since OS0 indicates that the operating system is not known. Still, the model does not predict the OSSCAN action because it is strongly negatively influenced by SERVICE0 and SERVICESSCAN, which reduce its confidence in selecting this action.

In a context different from the one previously presented, manual intervention analyses confirm that unsetting the OS or service information (turning it back to OS0 or SERVICE0 respectively) changes the chosen action 100% of the time (as getting that information is now back to being a priority).

C. Param selection

In this section, we analyze how discovered service names influence our model’s parameter selection. To illustrate this, we constructed a specific scenario where the host uses OS2 and SERVICE2, with one available exploit specifically requiring these two features.

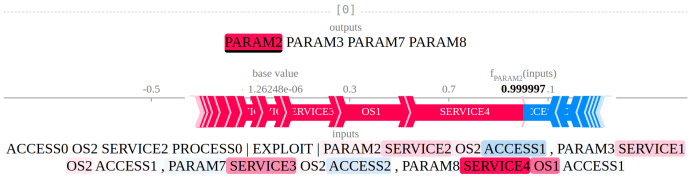


Fig. 7. Shap value of label ”PARAM2”

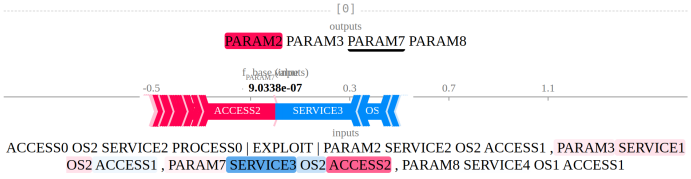


Fig. 8. Shap value of label ”PARAM7”

When examining the SHAP values for the PARAM2 prediction in Figure 7, we observe that contributions come from various features. Several unrelated elements (e.g., SERVICE4, OS1, SERVICE3) contribute positively to the prediction, whereas SERVICE2, which actually matches the service running on the target machine, is only the fourth most positively weighted feature.

To understand how the model still selects the correct parameter, we examine how it evaluates alternative parameters. Each alternative receives strong negative contributions from incompatible services. For example, Figure 8 illustrates a negative contribution from SERVICE3 when predicting PARAM7, which is expected since this parameter does not include that service. Similar patterns are observed for PARAM3 and PARAM8. We hypothesize that rather than directly selecting compatible features, the model eliminates incorrect options by identifying contradictions, ultimately choosing the only parameter that does not contradict the machine context.

D. Discussing BERT vs zero-shot

In our previous work on the zero-shot model [2], we computed SHAP values and observed that the model was often influenced by irrelevant or nonsensical tokens (such as ‘the’, ‘.’, etc.). This observation was one of the motivations for restricting the input context in our current work, where we provide only direct information that could meaningfully support the model’s decision-making process. This approach enables more logical reasoning that aligns with the input context.

VI. Live performance measure

In this section, we present the pipeline used for decision-making in the RL environment, the evaluation metric used to compare different RL agents, and the performance of our RedTeamBERT model compared to the zero-shot and CLAP models across four scenarios.

A. Sequential BERTs for decision-making

Figure 9 depicts the complete RedTeamBERT architecture during the evaluation phase, which combines the three specialized models described in the previous section. The decision-making pipeline operates sequentially: it first selects the target machine to attack, then chooses the appropriate action to perform on it, and finally, if the action requires parameters, it selects the most suitable ones to execute the action on the chosen machine. The contextual information used at each decision level is detailed in the previous section. Additionally, the pipeline includes two mechanisms designed to handle various network topologies: the ”padding system” and the ”roll system”.

1) Padding system: The padding system was developed after we observed that our RedTeamBERT model could be influenced by the length of the input context (either the machine context or the parameter context). To mitigate this issue, we introduced a padding system for both machine and parameter selection. The padding system is generally used to handle topologies that are smaller than those encountered during the training phase. It is triggered whenever the number of machines or parameters available at inference time is smaller than the number observed during training. For

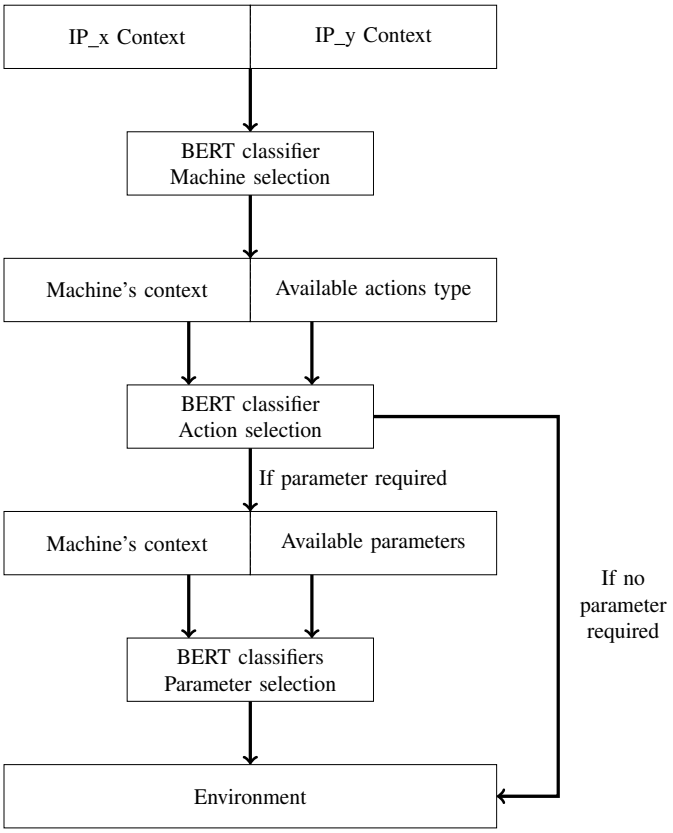


Fig. 9. Decision path taken by our model in inference mode

example, if the model was consistently trained on 4 parameters but encounters only 3 during inference, we automatically add dummy parameters to preserve the expected total of 4. This guarantees fixed-length input sequences, enabling stable decision-making by RedTeamBERT. We chose to fix the context size to 4 machines for machine selection, and 4 parameters for parameter selection. These values correspond to the maximum context sizes observed during training. This mechanism prevents the model from making suboptimal decisions due to context length mismatches between training and evaluation.

2) Roll system: The roll system was developed to enable the BERT model to handle topologies larger than those encountered during the training phase. It is triggered whenever the number of machines or parameters available at inference time exceeds the number seen during training. The roll system operates by selecting a pool of N machines (or parameters) and making an initial prediction. It then creates a new pool by retaining the top prediction and replacing the remaining N-1 items with new candidates. The model processes this new pool to generate another prediction, and this rolling process continues until all machines (or parameters) have been evaluated. In the end, we obtain a prediction that reflects the best machine (or parameter) across the entire set. In theory, this approach creates an unfair advantage for parameters appearing later in the list, since those evaluated early must survive multiple rounds where they could be incorrectly eliminated because of repeated model predictions. However, that is not a problem in our case, because we demonstrated that our BERT-based models for both machine and parameter selection achieve 100% accuracy.

The padding system and roll system can be complementary and may be triggered simultaneously. This can occur, for instance, when the topology is smaller than in the training examples but requires more exploit parameters than expected. Conversely, it may also happen when the topology is larger, but fewer parameters are available compared to those observed during training.

B. Evaluation metric

The standard performance metric for evaluating RL agents on the NASim environment is typically based on the number of actions required to compromise the target machine. However, we argue that this metric is suboptimal for evaluation purposes, as it penalizes exploratory actions aimed at gathering information about network hosts, treating them as unnecessary overhead, and instead favors direct exploitation attempts. Such behavior encourages rote learning and diverges significantly from that of a human pentester, who generally seeks to collect relevant information about a target before initiating any attack, especially in unfamiliar network environments. To address this limitation, we reuse the evaluation metric developed in our previous work [2] that explicitly encourages information-gathering behavior prior to exploitation, thereby aligning the agent's decision-making process more closely with real-world pentesting practices.

$$R_{final} = R_{standard} + \alpha * \sum_{t=1}^{n} \mathbf{1}_{\text{coh}}(a_t, c_t) \tag{1}$$

Our evaluation metric returns the classic reward plus a bonus at the end of the challenge. Equation 1 formalizes this metric, which provides a more accurate assessment of RL agent performance during the evaluation phase. The classic reward, denoted as $R_{standard}$, corresponds to the cumulative cost of all actions performed by the agent. The bonus term is defined as the amount of all actions a_t that are coherent when executed within their respective context c_t, weighted by a configurable factor α. The coherence function $\mathbf{1}_{\text{coh}} : A \times C \rightarrow \{0, 1\}$ returns 1 if all the information required for normally executing action a_t is displayed within context c_t, and returns 0 otherwise. The coefficient α compensates for the cost of information-gathering actions (e.g., os_scan, service_scan, process_scan). Agents that leverage discovery actions to guide their exploits receive a bonus reward. This factor must be set higher than 3, as compromising a machine typically requires discovering at least three attributes: operating system, services, and processes, each of which incurs a cost of -1. The metric is applied consistently across all evaluated agents (CLAP, zero-shot, and RedTeamBERT), and includes a safeguard to prevent multiple bonuses for repeated discovery of the same information.

C. Results

To evaluate our RedTeamBERT model against baselines, we used four scenarios: the tiny and small-linear scenarios already studied in our previous work with the zero-shot model, and two new scenarios, medium-test and large-test, inspired by the original medium and large scenarios. Table IV summarizes the network topology used in each scenario. We do not compare our work against PentestGPT, as the latter is not fully autonomous.

TABLE IV
NETWORK SCENARIOS USED BY OUR MODEL.

	Subnets	Hosts	OS	Services	Processes
Tiny	3	3	1	1	1
Small-linear	6	8	2	3	2
Medium-test	5	15	2	5	3
Large-test	10	27	2	5	3

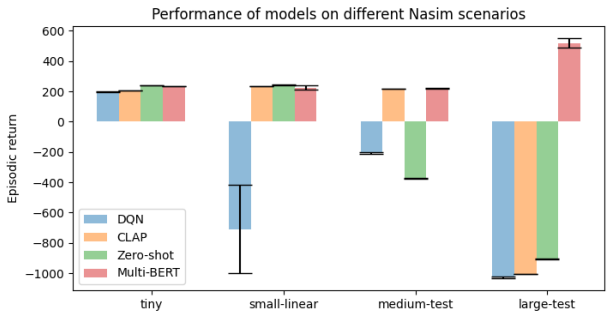


Fig. 10. Performance comparison of RL agents (DQN, CLAP, zero-shot, and RedTeamBERT) trained and evaluated with the new evaluation metric

Figure 10 shows the performance of our RedTeamBERT model compared to other RL agents (DQN, CLAP and zero-shot). We compare them across the four scenarios, using the mean episodic return computed over five runs. The episodic return represents the total reward accumulated by the agent throughout an episode, computed using the new evaluation metric introduced in the previous section. Note that the DQN and CLAP models are always trained on the scenarios on which they are tested, our model was trained exclusively on separate scenarios with four machines, and the zero-shot model is not trained on any scenario.

As shown in Figure 10, the DQN only figures out the path for the tiny scenario, the zero-shot model for the first two scenarios, and CLAP demonstrates strong performance on the tiny, small and medium scenarios only. Each time, these architectures fail when the scenario gets larger.

In addition, both DQN and CLAP are non-adaptive and are tested on the same network they were trained on. Therefore, we actually show the results of different DQN and CLAP models, each one trained on the exact scenario it is tested on. By contrast, our RedTeamBERT approach is trained on a database extracted from networks that are distinct from those used during testing. Therefore, it is the only architecture that is generalizable.

Performance-wise, our model's performance is comparable to that of CLAP and zero-shot for tiny and small scenarios, it is slightly better than CLAP for the medium scenario, and it is much better than the both of them for the large one, as it is the only one that manages to finish it.

TABLE V
RESULTS ON BIGGER SCENARIOS (5 TO 10 SUBNETS).

	DQN	CLAP	Zero-shot	Multi-BERT
Success ratio (at least once)	60%	80%	9%	100%
Success ratio (total)	39%	63%	5.5%	93.5%

We also conducted additional experiments on random scenario batches to compare the generalization abilities of each algorithm. We evaluate models on 100 random medium scenarios (5 subnets, 15 to 20 machines) and 100 random large scenarios (10 subnets, 20 to 40 machines). We can observe in table V that our RedTeamBERT model achieves a much better success rate compared to DQN, CLAP and zero-shot. It does not succeed every time however. The reason behind this is that NASim exploits are made to fail randomly with a certain percentage. Therefore, an attack may either fail because the firewall between the machines makes them impossible, or because we were unlucky when attacking. Both situations are indistinguishable during simulation, which explains the 6.5% of runs that fail for our system.

VII. CONCLUSION

In this paper, we presented a robust and adaptive approach based on four BERT classifiers that outperforms another RL agent (CLAP) as well as our previous work (zero-shot) across different scenarios in the NASim environment, according to our evaluation metrics. In some case, our BERT doesn't reach the maximum episodic reward, however it generally obtains a better episodic reward compared to the other RL agents. We demonstrated a certain level of explainability, which we consider to be superior to that of the previous zero-shot approach.

During some experiments, we observed that our BERT-based pipeline was not always able to complete the challenge, as it failed to learn certain conditions present in random scenarios. In future works, we plan to improve our pipeline to address this limitation and enable it to handle all random scenarios. We also aim to extend our BERT architecture to allow direct training within the RL environment, removing the need to generate a static dataset beforehand. Additionally, we plan to test our pipeline on a real cyber range to assess its performance in practical settings. Finally, we will continue to study the explainability of our model, as we consider this aspect crucial for deployment in real-world applications.

Our version of NASim, our dataset, and our model will be available at the following URL: https://github.com/silicom-hub/bert-pentesting-paper.

REFERENCES

[1] Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. PentestGPT: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 847–864, Philadelphia, PA, August 2024. USENIX Association.

[2] Christophe Genevey-Metat, Dorian Bachelot, Tudy Gourmelen, Adrien Quemat, Pierre-Marie Satre, Loïc Scotto, Di Perrotolo, Maximilien Chaux, Pierre Delesques, and Olivier Gesny. Red Team LLM: towards an adaptive and robust automation solution. In *Conference on Artificial Intelligence for Defense*, Rennes, France, November 2023. DGA Maîtrise de l'Information.

[3] Kento Hasegawa, Seira Hidano, and Kazuhide Fukushima. Autored: Automating red team assessment via strategic thinking using reinforcement learning. In *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy*, CODASPY '24, page 325–336, New York, NY, USA, 2024. Association for Computing Machinery.

[4] Junjie Huang and Quanyan Zhu. Penheal: A two-stage llm framework for automated pentesting and optimal remediation, 2024.

[5] Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. Nasimemu: Network attack simulator & emulator for training agents generalizing to novel scenarios, 2023.

[6] Minjune Kim, Jeff Wang, Kristen Moore, Diksha Goel, Derui Wang, Ahmad Mohsin, Ahmed Ibrahim, Robin Doss, Seyit Camtepe, and Helge Janicke. Cyberally: Leveraging llms and knowledge graphs to empower cyber defenders, 2025.

[7] Jakob Nyberg and Pontus Johnson. Structural generalization in autonomous cyber incident response with message-passing neural networks and reinforcement learning, 2024.

[8] Jonathon Schwartz and Hanna Kurniawati. Autonomous penetration testing using reinforcement learning. *CoRR*, abs/1905.05965, 2019.

[9] Yizhou Yang, Mengxuan Chen, Haohuan Fu, and Xin Liu. Settron: Towards better generalisation in penetration testing with reinforcement learning. In *IEEE Global Communications Conference, GLOBECOM 2023, Kuala Lumpur, Malaysia, December 4-8, 2023*, pages 4662–4667. IEEE, 2023.

[10] Yizhou Yang and Xin Liu. Behaviour-diverse automatic penetration testing: A curiosity-driven multi-objective deep reinforcement learning approach, 2022.

[11] Shicheng Zhou, Jingju Liu, Yuliang Lu, Jiahai Yang, Yue Zhang, and Jie Chen. Mind the gap: Towards generalizable autonomous penetration testing via domain randomization and meta-reinforcement learning, 2025.

Plan Generation for Multi-Robot Missions Requiring Active Operator Involvement

Emile Siboulet,[12] Arthur Bit-Monnot,[1] Marc-Emmanuel Coupvent des Graviers,[2] Jacques Yelloz,[3] Christophe Guettier,[2] Simon Lacroix[1]

[1] LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
{emile.siboulet, arthur.bit-monnot, simon.lacroix}@laas.fr
[2] Safran Electronics & Defense, Massy, France
{emile.siboulet, marc-emmanuel.des-graviers, christophe.guettier}@safrangroup.com
[3] Safran Tech, Châteaufort, France
jacques.yelloz@safrangroup.com

Abstract

This paper extends a constraint-based planning approach to deal with mixed-initiative for complex multi-robot missions. Operators in the loop with multi-robot systems may have to interact intensively: explicitly considering their cognitive load while planning the missions is a critical problem to address. The purpose of this work is to take into account at mission planning time the operator capacity to supervise the mission execution, to ensure efficient and safe operations. We introduce new mental load related metrics in an automatic constraint-based planner. The optimization of these metrics yields better quality plans for the operators to supervise and interact during execution. The planning feasibility and performances are evaluated on realistic scenarios.

Introduction

Mixed-Initiative Planning and Acting

Mixed-initiative, as described by Jiang and Arkin (2015), refers to collaborative frameworks in which both human operators and robotic agents possess the autonomy to initiate, modify, or discontinue tasks based on a dynamic assessment of the situational context. It improves operational efficiency, ensuring that both human and robotic agents can proactively contribute to achieving common objectives. For instance, mixed-initiative can be used with purely reactive systems that simply suggest to the user the best choice to make. It can also be part of a deliberative approach to mission planning and execution.

In planning activities, mixed-initiative refers to an approach in which humans and automated planning systems contribute actively to the creation or modification of plans. Examples are the SHERPA project (Bevacqua et al. 2015), where an operator find plans with the help of an exploration system to search for missing people in the mountains, or in the context of Mars exploration (Bresina et al. 2004). Mixed-initiative planning leverages the strengths of humans and computers: humans provide contextual understanding, creative problem-solving and flexibility, while automated systems offer speed, consistency, and the ability to efficiently consider complex constrained situations or large volumes of information.

Beyond planning, the principles of mixed-initiative also apply to mission supervision and execution (Dixon, Wickens, and Chang 2005; Cummings and Guerlain 2007; Wilkins, Lee, and Berry 2003). Planning aims at harnessing the complementary capabilities of humans and robots, thereby enhancing performance, efficiency and safety in the proper achievement of the mission. It primarily allows to quickly repair jobs without jeopardizing the overall plan, and also to handle collaborative human/robot plan repair or even replanning when needed.

For both planning and supervision activities, a key aspect in mixed-initiative approaches is to endow the operator with the ability to properly interact with the planner and the robots. Apart from adequate ergonomics, this calls for information sharing, control algorithms, and interaction protocols. In addition, and especially for mixed-initiative execution supervision, the operator must be in a mental state that allows its intervention.

Two important notions come into play: the Situation Awareness (SA) and the NASA Task Load indeX (NASA-TLX) (Ruff, Narayanan, and Draper 2002). SA involves understanding the current state of the environment and the robots, interpreting data to project future states, so as to make decisions that align with the robots' capacities and goals. The NASA-TLX is a widely used workload assessment tool that evaluates the perceived workload experienced by individuals performing tasks. It provides a multidimensional rating system that encompasses different aspects of workload to capture a comprehensive view of the task demands placed on an individual. These dimensions include mental demand, physical demand, temporal demand, performance, effort and frustration. The NASA-TLX is a valuable tool in the development and evaluation of mixed-initiative systems, offering insights into how automated systems impact human operators. It should be noted that the analysis of the mental load of an operator on the field is an active research topic in the cognitive models community (Kokar and

Endsley 2012; Hollands, Spivak, and Kramkowski 2019; Endsley, Garland et al. 2000). These are still to be adapted to a decision support context such as ours.

Contributions

In this paper, we propose a method to plan complex multi-robot missions that explicitly consider the operator constraints, aiming at making possible the mixed initiative supervision of the plan execution. This is done by introducing operator-related metrics that produce plans which do not overwhelm the operator during execution monitoring. Considered missions involve a team of aerial and ground robots that must ensure a progression throughout a terrain structured as a navigation graph. A mission involves the accomplishment of specific tasks at a number of nodes, some of which are constrained relative to each other. Main contributions of the paper are

- The definition of the mission with a Constraint Satisfaction Problem (CSP) based planning models,
- the introduction of metrics to quantify the operator's operational workload and information gain, and
- the evaluation of the method on a realistic mission use case.

Outline

In the next section, we take a brief look at the literature on methods of interaction between an operator and a team of robots, and the ways in which they can be quantified. The two following sections state the considered problem, formalize it and present the operational planning model. Metrics of operator mental load and informational gain are then introduced, and mission planning results are presented, on a simple use case first, and then for a full-scale scenario.

Problem Statement

General Problem

We address the challenge of planning missions for heterogeneous robot teams tasked with traversing a designated area under specific constraints and auxiliary objectives. The scenario involves a group of robots operating autonomously in a specified zone, supported by a remote operator situated nearby to the area. The operator possesses the capability to teleoperate the robots. He is familiar with these kinds of missions and provides strategic support during the mission. The heterogeneity of the robot team is a critical aspect, as certain tasks within the mission can only be executed by specific robots, necessitating task allocation among the team members.

The exploration zone presents its own set of contingencies related to terrain, which may obstruct the execution of a pre-defined plan and require operator advice to achieve mission objectives. Furthermore, the robots face the possibility of losing functionality either through the depletion of resources or mechanical failures, introducing a layer of unpredictability that the mission planning process must account for. Such deviations from the initial plan can be detected by an experienced operator, who might take extra steps to modify the plan before problems arise. While telecommunication issues are present in these scenarios, they are not deemed significant enough to impede the ability to teleoperate the robots effectively and will not be considered into the planning model.

A pivotal element of the mission is the execution of durative tasks, which may necessitate the completion of prerequisite tasks or the simultaneous execution of multiple tasks. Another objective of the mission is to respect some temporal exclusion in designated areas, in which certain robots are expected to be outside for a given time.

Role of the Operator

During the mission, the operator's responsibilities encompass the verification of task completion, the maintenance of a comprehensive understanding of the mission's progress, and ensuring the safety of all involved agents. This multifaceted role requires the operator to continuously monitor task execution while ensuring they are performed accurately and efficiently.

In situations where multiple task execution paths are available, the operator must take informed decisions, prioritizing the safety of the agents while considering the mission's objectives and the current situational context. To be able to take the proper decisions, he must have a high level of situational awareness to adapt to dynamic mission environments. He must therefore possess a holistic view of the mission's advancement, integrating information from various sources to form a coherent picture of the current state and foresee potential issues that may arise. Hence, besides the consideration of the operator's mental load during the mission planning process, ensuring he is aware of the situation to be able to take decisions is also an important concern.

Problem Formalization

Let us consider a set of nr robots navigating directed graphs, each denoted by $G_i = (V, E_i)$, where i serves as the index for individual robots. In this context, $e_{ik} \in E_i$ represents the k^{th} edge available to robot i, with $v, v' \in V^2$ forming edges as $e_{ik} = (v, v')$. The traversal time for the k^{th} edge by robot i is denoted by a constant Tt_{ik}.

Each robot i have specific entry and exit points associated with respective vertices.

Let v, v' and v'' denote respectively entry, exit and any other random vertex. v and v' are indirectly identified by a variable B_{iv} defined for every vertex as: $B_{iv} = 1$, $B_{iv'} = -1$ and $B_{iv''} = 0$.

We assume a discrete time representation as natural numbers, where 0 is the initial time and $tmax$ denotes the planning horizon, at which all robots must have reached their exit nodes.

The mission incorporates a total of nt tasks, each needing to be performed at a specific position $T_m \in V$ where m indexes each task. Each task duration is specified by $Dt_m \in [\![0, tmax]\!]$. Each task is associated with a valid initiation window, denoted as $Wt_m \subseteq [\![0, tmax]\!]$, ensuring that tasks are started within specified time slots. Additionally, particu-

lar tasks are required to be executed by specific robots, these assignments are detailed in the set $At_m \subseteq [\![1, nr]\!]$.

In parallel, the mission features nra temporal restrictions, represented as $Ar_n \in V$, where n indexes each temporal exclusion. The implication of a temporal exclusion varies, impacting only a subset of robots. Robots that must respect temporal exclusions are identified in the set $Mar_n \subseteq [\![1, nr]\!]$ for the n^{th} area. The position must not be occupied by robots that have to comply with the temporal exclusion during a time window $War_n \subseteq [\![0, tmax]\!]$.

Coordination between task execution is categorized into two types to streamline the framework, focusing on pairwise interactions. The sets S and P encompass action pairs that require synchronous and successive execution, respectively.

Furthermore, the mission strategy includes deploying agents for recognition. The corresponding boolean constant Af_{iv} is true iff the robot i is allowed to arrive first at position v.

The assignment must be carried out within a predefined timeframe, but the plan quality will be assessed in relation to the speed of its planned execution.

Planning Problem Formulation

In this section, we propose an encoding of the problem into a CSP formalism with finite-domain integer variables, inspired by the one of Guettier (2007) on related progression problems. This approach is particularly efficient for producing plans. Boolean variables are represented as binary integer variables where 1 encodes *true* and 0 encodes *false*. We denote disjunctions by vertical lines to the left of the equation. All constants are denoted by non-qualigraphic letters. Variables are denoted by qualigraphic letters.

Navigation Graph

The planning of the actions carried out by the agents is done using a flow model. We define $\mathcal{F}_{ik} \in [\![0, 1]\!]$ the flow representing the path of the robot i on the k^{th} edge. Thus, ensuring flow consistency is equivalent to compare incoming, outgoing and balance flow. To do so, for each position, we have to constraint incoming and outgoing flow to be equal to balances.

$$\sum_{k|e_{ik}=(v',v)} \mathcal{F}_{ik} - \sum_{k|e_{ik}=(v,v'')} \mathcal{F}_{ik} = B_{iv} \tag{1}$$

Propagation of Operational Metrics

The primary objective of this mission planning model is to produce schedules for the robot actions. It is therefore necessary to have a time metric based on the robots' achievements. To do this, we have chosen to represent time with two variables $\mathcal{T}_{iv}, \mathcal{D}_{iv} \in [\![0, tmax]\!]^2$ that represent the time of arrival and the duration of the time spent on the node v of the robot i. It has to be propagated on the graph as:

$$\mathcal{T}_{iv} = \sum_{k|e_{ik}=(v',v)} \mathcal{F}_{ik}(\mathcal{T}_{iv'} + \mathcal{D}_{iv'} + Tt_{ik}) \tag{2}$$

By the network flow definition, $\mathcal{T}_{iv}$ is 0 where the agent does not use the position v. We need to add a constraint on $\mathcal{D}_{iv}$ so it's also 0 where agent does not pass. This is the case for every point where $\mathcal{T}_{iv}$ is 0 except the entry point. As B_{iv} represent flow bias it could be used in the logic equation to represent the entry point where $B_{iv} = 1$. Thus the constraint asserting null duration on none pass by positions.

$$\mathcal{T}_{iv} = 0 \wedge B_{iv} \neq 1 \Rightarrow \mathcal{D}_{iv} = 0 \tag{3}$$

Task Constraint Expression

Considering the task m, we designate $\mathcal{R}t_{im} \in [\![0, 1]\!]$ that reify task completion by the agent i. We first add the constraint that it has to be completed by one agent that is allowed to do so.

$$\sum_i \mathcal{R}t_{im} = 1 \tag{4}$$

$$i \notin At_m \Rightarrow Rt_{im} = 0 \tag{5}$$

In our planning model, we consider the task to be completed if the robot remains on the point longer than the duration of the task to be carried out. Thus durative task realization constraint is

$$\mathcal{R}t_{im} \Rightarrow \mathcal{D}_{iT_m} \geq Dt_m \tag{6}$$

We define the variable $\mathcal{T}t_m \in [\![0, tmax]\!]$ the starting time of realization of the task m. This variable will be used later for verification of synchronization. In addition to the storage of the time of the realization, we have to ensure that this is performed in the appropriate time window.

$$\mathcal{R}t_{im} \Rightarrow \mathcal{T}t_m = \mathcal{T}_{iT_m} \wedge \mathcal{T}_{iT_m} \in Wt_m \tag{7}$$

We also need to consider the limiting case of the starting point. The robot is at the starting position at t=0. This is also the case for all positions not taken by the robot. It is therefore necessary to make the task feasible at t=0 for the starting point, or to ensure that it is carried out at a later time to ensure the robot's passage.

$$\mathcal{R}t_{im} \Rightarrow \mathcal{T}_{iT_m} \geq 1 \vee B_{iT_m} = 1 \tag{8}$$

Tasks cannot be performed simultaneously by the same agent at the same location, therefore, we prohibit an agent from executing two tasks at the same node.

$$m \neq m' \wedge \mathcal{T}t_m = \mathcal{T}t_{m'} \Rightarrow \mathcal{R}t_{im} + \mathcal{R}t_{im'} \leq 1 \tag{9}$$

Temporal Exclusion Constraint

Temporal exclusion constraints on a given position is expressed by the following disjunction. The robot must pass before or after the temporal exclusion window War_n.

$$i \in Mar_n \Rightarrow \mathcal{T}_{iAr_n} + \mathcal{D}_{iAr_n} < War_n \vee \mathcal{T}_{iAr_n} > War_n \tag{10}$$

Position Discovery

To ensure the constraint of first arrival it is necessary to define two new variables $\mathcal{T}f_v, \mathcal{I}f_{iv} \in [\![0, tmax]\!] \times \mathbb{B}$ which respectively represent the time of the first robot's arrival time and whether i is the first robot arriving at position v.

We state the disjunction between three cases. Either the robot is the first to arrive at the position, and the moment of first arrival is when the robot enters the position. Or it passes over the position and is not the first, hence the time of first arrival is lower than the robot's arrival time. Or it does not pass through this position.

$$\left| \begin{array}{l} \mathcal{I}f_{iv} \wedge \mathcal{T}_{iv} = \mathcal{T}f_v \\ \neg\mathcal{I}f_{iv} \wedge \mathcal{T}_{iv} > \mathcal{T}f_v \\ \neg\mathcal{I}f_{iv} \wedge \mathcal{T}_{iv} = 0 \end{array} \right. \tag{11}$$

It is therefore only possible for an agent to arrive alone on a position first. It is also necessary to force the use of these variables if a robot passes through this position.

$$\sum_i \mathcal{T}_{iv} \geq 1 \Rightarrow \sum_i \mathcal{I}f_{iv} = 1 \tag{12}$$

It is also necessary to add constraints on position never visited. We need to enforce that if no robot passes by the position then there is no first arrived at this position.

$$\sum_i \mathcal{T}_{iv} = 0 \Rightarrow \sum_i \mathcal{I}f_{iv} = 0 \tag{13}$$

Coordination Between Tasks

For the succession of 2 tasks $(m, m') \in P$ we want to ensure that the first task is completed before starting the second one. Thus the constraint for every pair of successive tasks m and m'

$$\mathcal{T}t_m + \mathcal{D}t_m \leq \mathcal{T}t_{m'} \tag{14}$$

For the synchronization of 2 tasks $(m, m') \in S$ we want to ensure that both tasks start at the same time. Thus the constraint

$$\mathcal{T}t_m = \mathcal{T}t_{m'} \tag{15}$$

Mission Metrics

We want the robots to arrive as soon as possible at the end of the mission thus reducing the mission's makespan. The makespan takes into account moving and task completion. It may be necessary to leave robots possibility of staying on the finish line if a task needs to be completed at the mission exit point.

$$\mathcal{M}m = \max_{iv}(\mathcal{T}_{iv} + \mathcal{D}_{iv}) \tag{16}$$

Resolution of the Constraint Problem

The constraint problem is solved using OrTools (Google LLC 2023). The model is expressed in Minizinc(min 2023; Nethercote et al. 2007).

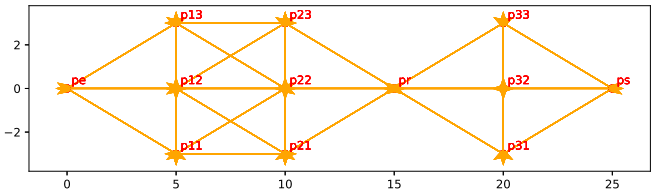


Figure 1: Test Navigation Graph

Problem Instance

We investigate planning outcomes on a demonstration scenario effects of the metrics introduced previously. The navigation graph for this example is shown in Figure 1.

The scenario includes five agents: two grounds unmanned vehicles (UGVs) traveling at 1 m/s and three unmanned aerial vehicles (UAVs) at 3 m/s. These agents adhere to a consistent navigation graph, entering at point *pe* and exiting at *ps*.

Throughout the mission, tasks are allocated at various numbered points, each taking 2s to complete. Some tasks have specific temporal requirements, such as simultaneous tasks at *p23* and *p11*, and at *p33* and *p31*. Moreover, the task at *p32* must precede the task at *p12*. Access to point pr is prohibited between 10s and 20s.

The optimal plan solution for this problem is shown in Figure 2a, with numerous tasks executed concurrently. *Go to position* tasks are displayed in blue and *durative* tasks in green. The resulting plan proposes a solution that satisfies all operational objectives. Temporal exclusions are represented in gray on a dedicated timeline below the agent, they are respected by every agent.

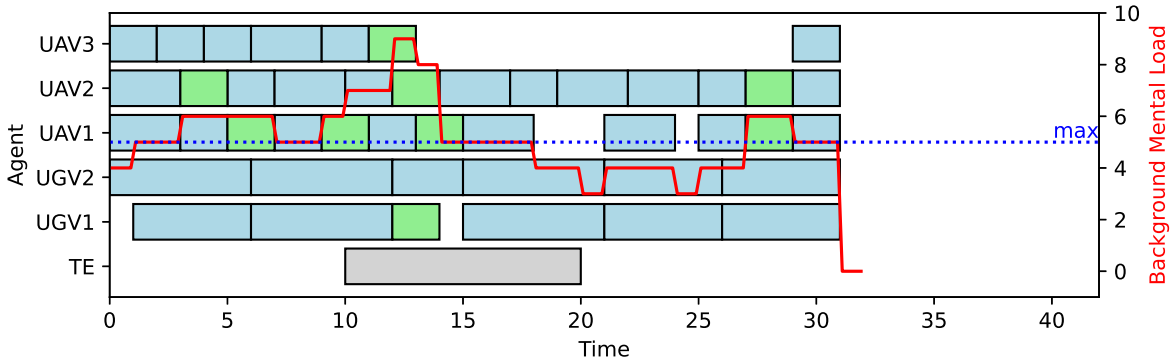
We represent with blue dot the maximum level of mental load desired that will be presented in the next section. The plan to minimize the makespan does not meet this criterion. It is necessary to define measurable mental load metrics, with associated desired threshold, to ensure effective operator mission supervision during execution.

Planning with Operator Based Metrics

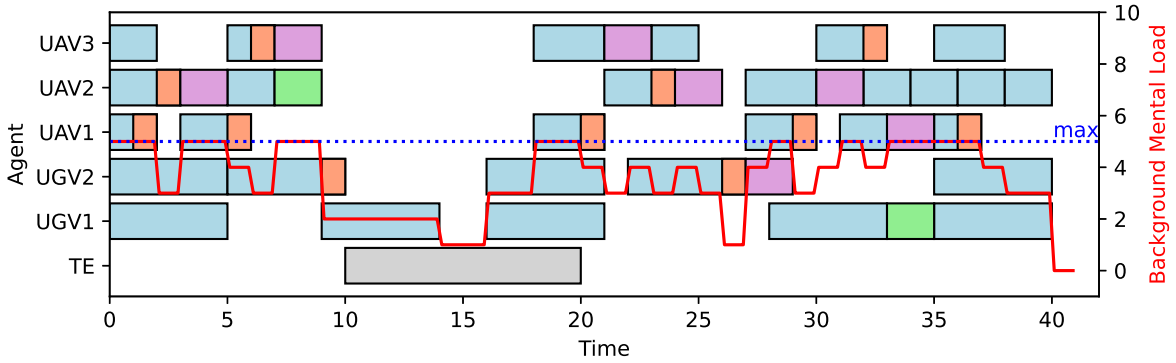
In our target scenarios, plan execution requires an operator responsible for overseeing operations, to guarantee the safety of the robots and efficiency of the mission. In this section we define operator based metrics that can be optimized to improve plan quality.

Metrics Overview

We propose three new metrics to optimize a plan along operator supervision. The metrics are inspired from ergonomics observations during real life experimentation such as described in (Ruff, Narayanan, and Draper 2002; Dixon, Wickens, and Chang 2005; Cummings and Guerlain 2007). We hypothesize that the automatic system is able of carrying out the mission in complete autonomy. The operator is responsible for ensuring robot safety and plan completion. To do this, it has robot positions and robots states and completion



(a) Makespan Optimization



(b) Operator Based Metrics Optimization

Figure 2: Temporal Representation of Task Completion

information in progress tasks. In addition, it can directly supervise the tasks to be carried out. For material reasons, we consider that only one task can be supervised at a time.

- The first metric deals with **tasks or temporal exclusion supervision** by the operator. Ideally, the operator should supervise as many tasks and exclusions as possible, with a preference for complex tasks. However, due to hardware constraints, an operator can only supervise one task or temporal exclusion at a time. The aim of this metric is primarily to secure task completion, with positive effects on operator situation awareness.
- The second metric represents **situational knowledge acquisition**. Some terrain key positions provide important insights into mission status, and it is important to explore them regardless of task completion. While the operator supervises the robots, discovering some key positions increase operator awareness levels. We define a new task that doesn't correspond to any operational action in the mission. Supervision on position discovery consists of a brief consideration of robot perception. It is necessary to supervise as many positions discoveries as possible, preferably those who are the more interesting. The aim of this metric is to improve the situation awareness of the operator.
- The third metric represents the **operator background mental load**. During a mission, the operator is constantly trying to keep in mind what robots are doing. This is particularly problematic during critical operations when the operator would like to focus on a single robot. When the operator is overcommitted by the mission, the results are even worse. In fact, instead of providing a decision support, robots are abandoned until the critical task is resolved. Constraints over the metric aim to decrease task load index.

Formalization

Regarding background mental load metrics, we associate to each task a scalar $L_a \in \mathbb{N}$, for each action $a \in [\![1, nt + nar]\!]$. Each robot performing a *move to a position* task is counted as 1. The background mental load metric maximum

is defined by $Mlmax \in \mathbb{N}$.

The informational gained by the discovery of the v^{th} position is specified in $G_v \in \mathbb{N}$. When the discovery is made through a supervised action, we consider that the information gain to be scaled by a factor Ig.

Supervised Action

We define an action as a task or a temporal exclusion. To represent the operator's supervised action, we define the boolean variable $\mathcal{S}_a \in \mathbb{B}$ which is true iff the action is supervised. Due to hardware constraints, only a single action supervision is possible at any point in time. This constraint is asserted on every pair of tasks m, m'. So either some tasks are not supervised, or their execution times do not overlap, which is enforced by the disjunctive constraint.

$$\left| \begin{array}{l} \mathcal{S}_m + \mathcal{S}_{m'} \leq 1 \\ \mathcal{T}t_m + Dt_m < \mathcal{T}t_{m'} \\ \mathcal{T}t_{m'} + Dt_{m'} < \mathcal{T}t_m \end{array} \right. \tag{17}$$

The same need to be done for temporal exclusions but on their time windows.

$$\left| \begin{array}{l} \mathcal{S}_n + \mathcal{S}_{n'} \leq 1 \\ War_n < War_{n'} \\ War_{n'} < War_n \end{array} \right. \tag{18}$$

Cross concurrency also needs to be addressed by combining previous approach.

$$\left| \begin{array}{l} \mathcal{S}_m + \mathcal{S}_n \leq 1 \\ War_n < \mathcal{T}t_m \\ \mathcal{T}t_m + Dt_m < War_n \end{array} \right. \tag{19}$$

The metric representing task and temporal exclusion supervision is obtained by adding the performed supervision weighted by their relevance.

$$\mathcal{M}s = \sum_m \mathcal{S}_m L_m + \sum_n \mathcal{S}_n L_n \tag{20}$$

Situational Knowledge Acquisition

We define the variable $\mathcal{F}s_{iv} \in \mathbb{B}$ representing the supervised discovery of the position v by the robot i. As it is concurrent with task supervision, we have to enforce temporal separation. As for tasks or temporal exclusion supervision, we first ensure that both are not done in the plan or they are not overlapping.

$$\left| \begin{array}{l} \neg(\mathcal{F}s_{iv} \wedge S_m) \\ \mathcal{T}_{iv} > \mathcal{T}t_m + Dt_m \\ \mathcal{T}_{iv} \leq \mathcal{T}t_m \end{array} \right. \tag{21}$$

A similar constraint is used to prevent overlapping supervised discovery and temporal exclusion supervision:

$$\left| \begin{array}{l} \neg(\mathcal{F}s_{iv} \wedge S_n) \\ \mathcal{T}_{iv} > Wra_n \\ \mathcal{T}_{iv} \leq Wra_n \end{array} \right. \tag{22}$$

It is only possible to perform one supervision at a time, so it is necessary to specify the disjunction of discovery supervision two by two.

$$\left| \begin{array}{l} \neg(\mathcal{F}s_{iv} \wedge \mathcal{F}s_{i'v'}) \\ \mathcal{T}_{iv} \neq \mathcal{T}_{i'v'} \end{array} \right. \tag{23}$$

The associated metric is denoted $\mathcal{M}fs$ and defined as follows.

$$\mathcal{M}fs = \sum_v G_v \sum_i \mathcal{I}f_{iv}[Ig\mathcal{F}s_{iv} + (1 - \mathcal{F}s_{iv})] \tag{24}$$

Background Mental Load

As described beforehand, background mental load is the action performed by each robot at a specific time. For a given time $t \in [\![1, tmax]\!]$, we define $actT(t)$ as the set of tasks executing at t:

$$actT(t) = \{m \in [\![1, nt]\!] \mid t \in [\![\mathcal{T}t_m, \mathcal{T}t_m + Dt_m - 1]\!]\} \tag{25}$$

$actAr(t)$ as the set of active temporal exclusions:

$$actAr(t) = \{n \in [\![1, nar]\!] \mid t \in Wra_n\} \tag{26}$$

and $move(t)$ as the set of ongoing displacement at t:

$$move(t) = \left\{ v \in [\![1, nr]\!] \;\middle|\; \begin{array}{l} t \in [\![\mathcal{T}_{iv}, \mathcal{T}_{iv} + \mathcal{D}_{iv} - 1]\!] \\ \vee\, t \geq \mathcal{T}_{iv} \wedge B_{iv} = -1 \end{array} \right\} \tag{27}$$

The background mental load $\mathcal{M}l(t)$ is modeled as the sum of the contribution of all these activities at a given time t:

$$\mathcal{M}l(t) = \sum_{m \in actT(t)} L_m + \sum_{n \in actAr(t)} L_n + \sum_{v \in move(t)} 1 \tag{28}$$

We do not optimize this metric but constraint its value below a certain threshold defined during mission preparation.

$$Mlmax \geq \max_t(\mathcal{M}l(t)) \tag{29}$$

Metrics optimization

While optimizing operator metrics, optimal makespan is no longer feasible. The cognitive load is not optimized, it is limited so as not to overload the operator such as described in equation 29. Optimization step has to be conducted iteratively to achieve the best quality plan. Optimal makespan value might not be reached.

1. Maximizing supervision score ($\mathcal{M}$),
2. Maximizing informational gain ($\mathcal{M}_{fs}$),
3. Minimizing makespan ($\mathcal{M}m$),
4. We have tie-breaking metric to that indirectly favors a fair repartition of tasks among agents.

Thus the global metrics to be maximized is of the shape

$$\mathcal{M} = A\mathcal{M}s + B\mathcal{M}fs - C\mathcal{M}m - \sum_{iv}(T_{iv}^2) \tag{30}$$

With constants A, B, C such that the resolution is lexicographic ($A \gg B \gg C \gg 1$).

Results

Problem's optimal solution is depicted in Figure 2b. In addition to Figure 2a color convention, we display supervised tasks in purple and supervised position discovery in orange.

Operator maximal background mental load criterion is respected which causes the makespan to be larger. Not all tasks are supervised, as some are performed synchronously. This synchronicity is a hard constraint of the planning problem. The final optimization step separates the different tasks and produces a plan that is more resilient to minor variations in execution time.

Application to a Realistic Operation

CoHoMa II Mission

French Army introduced a new challenge within a military context that involves navigating through hostile territories. This requires coordinating robots to ensure the safety of the operator's vehicle, as presented by Godet, Lesire, and Bit-Monnot (2023).

It involves a robotic combat group tasked with progressing 1.5 km in enemy territory while ensuring the protection of operators inside a command vehicle. Contingencies are simulated by disseminated red cubes, discover during the progression. Instructions on the cubes detail the operations required for their deactivation, which may require the collaboration of multiple robots. The vehicle is considered vulnerable and must avoid proximity to the red cubes unless they have been deactivated beforehand.

Model

To model the CoHoMa mission, we used distinct navigation graphs for UAVs and UGVs. For UGVs, their real geometry is obtained by representing the different paths in a forest. For UAVs, the navigation graph is obtained by connecting all nearby points which gives them greater freedom of trajectories. The speed settings are 1 meter per second for UGVs and 3 meters per second for UAVs.

We introduce a new agent in the planning model that represents the operator who has to cross the area. All reached position by this agent has to be discovered by another one. This agent is not able to be performed durative tasks and has a speed of 1 m/s.

The mission involves various entry and exit points for the agents, which are not further elaborated upon in this text. It involves achieving 13 tasks, each with the duration of 120 seconds. Two tasks have a specific order of execution and three tasks are on the same position and have to be synchronized. There is a temporal exclusion from t=200s to t=400s where no task has to be conducted. The makespan of the plan is required to be below 2500 seconds, with a time step of 20 seconds.

Results

On this representative problem, when ignoring operator metrics, optimal makespan is computed in 970s. An optimal solution with respect to operator metrics is displayed in Figure 3. This solution was computed in 1800s on a machine with 10 physical core of 2.6 GHz, with OrTools v9.8.

This plan, while being 120% longer to execute linken to the optimal plan, effectively reduces the peak of the operational workload by three which greatly facilitates comprehensive exploration of the zone. The strong impact of the max load on the makespan is due to the fact that there are more agents than the max load. This means that not all agents can move in parallel. In addition, the tasks modeled are particularly demanding, which makes it even more difficult for robots to move.

It should be noted that incorporating operator metrics into the constraint problem presents significant complexity for the solver. Indeed, it is easier for a CSP solver to take into account metrics correlated to the task progression such as the makespan or the fuel usage. On the other hand, operator metrics are orthogonal to the propagation of space progression since they depend on time. Consequently, scaling emerges as a critical and complex aspect to consider. Our effort has been toward an implementation that facilitates problem resolution. Implementing dedicated resolution strategies for these metrics could be a way of greatly optimizing solutions search. Precisely characterizing these tradeoffs between the model accuracy and the runtime of the solver will be the purpose of a more detailed empirical analysis on a more diverse set of problems.

Discussion

This method has not been tested during the CoHoMa II mission but was designed based on the feedback from two teams who participated in the challenge. During this challenge, it became apparent that the operators were unable to fulfill their critical roles when resorting to existing planning approaches without operator-specific metrics.

The proposed operator metrics significantly advance the usability of planning in this context, by better aligning the plan with the capabilities of the operator to supervise plan execution. This enables the direct involvement of the operator, e.g., by means of teleoperation, to deal with contingent events and situations not anticipated in the generated plan. Beside our own scenario, the need for such direct involvement of the operator at execution time is notably justified in a military context by (Dixon, Wickens, and Chang 2005; Wilkins, Lee, and Berry 2003; Cummings and Guerlain 2007).

Our approach to take the operator into account in planning considers a simplistic model of mental load. Finer cognitive and mental load models should be considered, such as the ones proposed for stressful situations involving high levels of responsibility in (Kokar and Endsley 2012; Hollands, Spivak, and Kramkowski 2019; Endsley, Garland et al. 2000).

But even with more elaborated models, a more fundamental limitation is the difficulty of precisely capturing the desires of the operator in the optimization metric. One could even advocate that this is impossible to do, as the role of the operator is precisely to bring a different perspective than what can be currently captured in a computer system. To tackle this challenge, we believe it is interesting to push forward mixed-initiative planning techniques. Mixed-initiative planning has been a subject of interest for many years, especially in the context of space operations (Ai-Chang

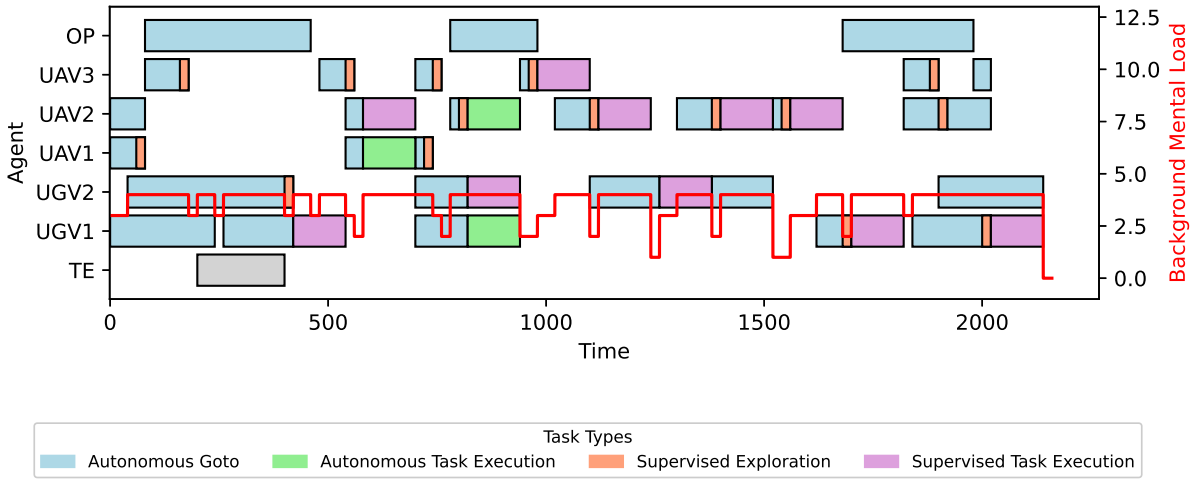


Figure 3: Time usage of different robots when taking account operators on CoHoMa mission

et al. 2004). A critical enabling feature for mixed-initiative planning is the ability for the automated planner to explain its decision to the operator. This has been the subject of recent work in both the automated planning (Eifler et al. 2020) and constraint programming (Guns et al. 2023; Gamba, Bogaerts, and Guns 2023) communities, which appear very relevant for our needs. Our interest is also the work of Lerouge et al. (2023), who considers the generation of explanations for flow-model similar to the one of Guettier (2007) and our own.

Besides, an underlying assumption of our approach is that, if a sufficient time is allocated for an action, the execution layer will successfully accomplish the task, possibly with the help of the operator. Should it fail nevertheless, we advocate for a strategy that involves replanning the mission considering the current state and any diminished capabilities.

Finally, several improvements could be brought to our planning model to tackle a more diverse set of problems. Notably, explicit representation of resources, such as fuel, would be important for a number of scenarios. Such resource constraints are very common in the constraint programming and operations research communities and would naturally fit in our formalism.

Conclusion

We have presented an operational planning model that is capable of modeling realistic planning problems as demonstrated on a complex progression mission. Its domain-dependent approach enables its efficient resolution by a constraint solver.

Through the introduction of operator-centric metrics, we presented contributions toward enabling operator's involvement at plan execution. A natural evolution of this work would be toward mixed-initiative planning, empowering the operator to refine the generated plan through direct interaction with the planning system. Typical interactions could be, *e.g.*, changing the priorities of optimization metrics or assigning a task to particular robots.

References

2023. MiniZinc. http://www.minizinc.org/. Accessed: 2024-03-12.

Ai-Chang, M.; Bresina, J.; Charest, L.; Chase, A.; Hsu, J.-J.; Jonsson, A.; Kanefsky, B.; Morris, P.; Rajan, K.; Yglesias, J.; Chafin, B.; Dias, W.; and Maldague, P. 2004. MAPGEN: mixed-initiative planning and scheduling for the Mars Exploration Rover mission. *IEEE Intelligent Systems*, 19(1): 8–12.

Bevacqua, G.; Cacace, J.; Finzi, A.; and Lippiello, V. 2015. Mixed-Initiative Planning and Execution for Multiple Drones in Search and Rescue Missions. *Proceedings of the International Conference on Automated Planning and Scheduling*, 25(1): 315–323.

Bresina, J. L.; Jónsson, A. K.; Morris, P. H.; and Rajan, K. 2004. Activity planning for the mars exploration rovers. In *ICAPS-2005 Conference*.

Cummings, M. L.; and Guerlain, S. 2007. Developing operator capacity estimates for supervisory control of autonomous vehicles. *Human factors*, 49(1): 1–15.

Dixon, S. R.; Wickens, C. D.; and Chang, D. 2005. Mission control of multiple unmanned aerial vehicles: A workload analysis. *Human factors*, 47(3): 479–487.

Eifler, R.; Cashmore, M.; Hoffmann, J.; Magazzeni, D.; and Steinmetz, M. 2020. A new approach to plan-space explanation: Analyzing plan-property dependencies in oversubscription planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9818–9826.

Endsley, M. R.; Garland, D. J.; et al. 2000. Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, 1(1): 3–21.

Gamba, E.; Bogaerts, B.; and Guns, T. 2023. Efficiently explaining CSPs with unsatisfiable subset optimization. *Journal of Artificial Intelligence Research*, 78: 709–746.

Godet, R.; Lesire, C.; and Bit-Monnot, A. 2023. Multi-Robot Task Planning to Secure Human Group Progress. *arXiv preprint arXiv:2310.07731*.

Google LLC. 2023. OR-Tools. https://developers.google.com/optimization/. Accessed: 2024-03-12.

Guettier, C. 2007. Solving planning and scheduling problems in network based operations. *Proceedings of Constraint Programming (CP)*.

Guns, T.; Gamba, E.; Mulamba, M.; Bleukx, I.; Berden, S.; and Pesa, M. 2023. Sudoku Assistant–an AI-Powered App to Help Solve Pen-And-Paper Sudokus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 16440–16442.

Hollands, J. G.; Spivak, T.; and Kramkowski, E. W. 2019. Cognitive load and situation awareness for soldiers: effects of message presentation rate and sensory modality. *Human factors*, 61(5): 763–773.

Jiang, S.; and Arkin, R. C. 2015. Mixed-initiative human-robot interaction: definition, taxonomy, and survey. In *2015 IEEE International conference on systems, man, and cybernetics*, 954–961. IEEE.

Kokar, M. M.; and Endsley, M. R. 2012. Situation awareness and cognitive modeling. *IEEE Intelligent Systems*, 27(3): 91–96.

Lerouge, M.; Gicquel, C.; Mousseau, V.; and Ouerdane, W. 2023. Counterfactual Explanations for Workforce Scheduling and Routing Problems. In *12th International Conference on Operations Research and Enterprise Systems*, 50–61. SCITEPRESS-Science and Technology Publications.

Nethercote, N.; Stuckey, P. J.; Becket, R.; Brand, S.; Duck, G. J.; and Tack, G. 2007. MiniZinc: Towards a Standard CP Modelling Language. In *Principles and Practice of Constraint Programming – CP 2007*, Lecture Notes in Computer Science, 529–543. Springer.

Ruff, H. A.; Narayanan, S.; and Draper, M. H. 2002. Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence*, 11(4): 335–351.

Wilkins, D. E.; Lee, T. J.; and Berry, P. 2003. Interactive execution monitoring of agent teams. *Journal of Artificial Intelligence Research*, 18: 217–261.

Agile Interception of a Flying Target using Competitive Reinforcement Learning

Timothée Gavin
IAS
Thales LAS
Rungis, France
timothee.gavin@thalesgroup.com

Simon Lacroix
RIS
LAAS CNRS
Toulouse, France
simon.lacroix@laas.fr

Murat Bronz
Dynamic Systems, OPTIM
Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse
Toulouse, France
murat.bronz@enac.fr

***Abstract*—This article presents a solution to intercept an agile drone by another agile drone carrying a catching net. We formulate the interception as a Competitive Reinforcement Learning problem, where the interceptor and the target drone are controlled by separate policies trained with Proximal Policy Optimization (PPO). We introduce a high-fidelity simulation environment that integrates a realistic quadrotor dynamics model and a low-level control architecture implemented in JAX, which allows for fast parallelized execution on GPUs. We train the agents using low-level control, collective thrust and body rates, to achieve agile flights both for the interceptor and the target. We compare the performance of the trained policies in terms of catch rate, time to catch, and crash rate, against common heuristic baselines and show that our solution outperforms these baselines for interception of agile targets. Finally, we demonstrate the performance of the trained policies in a scaled real-world scenario using agile drones inside an indoor flight arena.**

***Index Terms*—Reinforcement Learning, Multi-Agent Systems, Interception, Agile Flight**

I. Introduction

The interception of agile aerial targets using autonomous drones is a challenging and increasingly relevant problem in robotics and security. The increasing presence of unmanned aerial vehicles (UAVs) in unauthorized, restricted airspaces poses significant safety and security risks and has spurred interest in developing effective interception strategies [1] In particular, scenarios such as airspace protection, infrastructure security, and event safety require the ability to capture or neutralize unauthorized drones with high precision and minimal collateral risk. Deploying interceptor drones equipped with nets is a promising approach, but it demands advanced control capabilities to match or exceed the agility of evasive targets.

Traditional interception methods often rely on accurate models, preplanned strategies, or predictable target behaviour [2]. However, modern quadrotor drones can perform highly dynamic manoeuvres, and will actively evade capture, rendering their trajectories unpredictable and challenging the effectiveness of classical methods [3].

Recent advances in deep reinforcement learning (RL) have demonstrated the potential to learn complex, high-dimensional control policies for drones directly from interaction with the environment. In particular in drone racing, RL-trained policies have achieved superhuman performance in highly dynamic and agile flight tasks [4]. However, the drone racing problem typically involve navigating static or slowly moving gates, while interception requires reacting to an adversarial agent that actively attempts to evade capture.

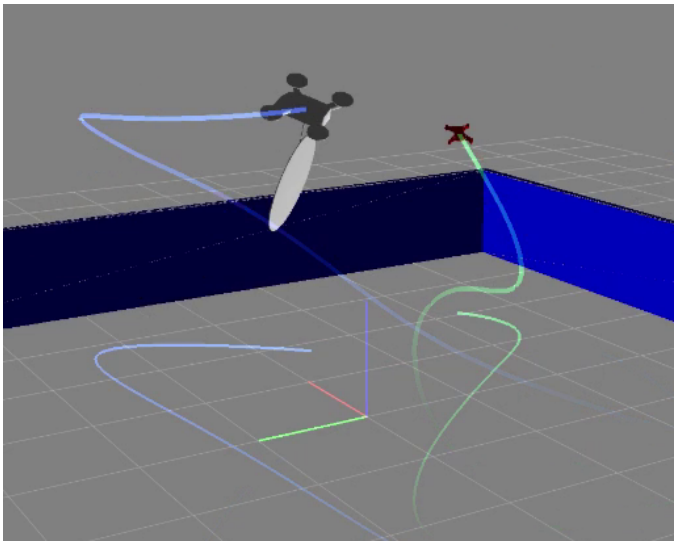


Fig. 1: A competitive reinforcement learning approach to train both a pursuer and an evader drone for agile interception tasks. Both agents learn low-level control policies that enable them to perform dynamic maneuvers in a high-fidelity simulation environment.

Competitive Multi-Agent RL (MARL) have shown outstanding results in adversarial settings, such as games [5], [6]. In this work, we formulate the agile interception problem as a competitive multi-agent RL task, where both the interceptor (pursuer) and the target (evader) are controlled by independent policies trained using Proximal Policy Optimization (PPO) in a co-evolution framework. Our approach integrates a high-fidelity quadrotor dynamics model, enabling both agents to learn agile, physically realistic manoeuvres from low-level control inputs. Through extensive simulation and real-world experiments, we show that RL training leads to robust and adaptive interception and evading strategies, outperforming heuristic control approaches.

The main contributions of this paper are:

- A competitive MARL framework for agile drone interception, with both pursuer and evader learning from low-level control.
- Integration of a realistic quadrotor dynamics model to enable physically realistic and agile flight behaviors.
- Empirical evaluation demonstrating superior performance over standard baselines in simulation.

The remainder of the paper is organized as follows. Section II reviews related work on interception and agile flight. Section IV and III detail our agile flight simulation environment and our training methodology. Section V presents experimental results and comparisons. Section VI and VII conclude and discuss future directions.

II. Related-Work on the Agile Interception problem

A. Agile flight

Agile flights in multi-rotors drones are typically characterized by the ability to perform large-angle manoeuvres, sustain high linear and angular accelerations, maintain precise control near dynamic limits and do so reliably in real-time, often in complex and cluttered environments. Traditionally, achieving such agility relied on trajectory optimization coupled with controllers like Model Predictive Control (MPC), often requiring pre-planned paths and accurate system models [7]. However, these methods can be brittle when faced with unexpected disturbances of real-world flights. Reinforcement Learning (RL) has emerged as a powerful alternative, enabling the learning of complex, non-linear control policies directly from interaction. Research, such as work from [4] and [8], has demonstrated RL's capability to achieve highly dynamic and agile flights for quadrotors, pushing the boundaries of autonomous aerial manoeuvring beyond what traditional methods could easily achieve, particularly in tasks requiring aggressive, near-limits flight.

B. Interception

Traditional interception methods uses heuristic or optimal control methods that often rely on accurate models, pre-planned strategies, or predictable target behaviour. [2]. These approaches have been historically designed for the control of missiles and the interception of fixed-wing manned aircraft. Optimal control methods requires accurate model of the pursuer and the evader to compute interception trajectories [9]. Such model may not be available or may be too computationally expensive for real-time adaptation against unpredictable targets. Heuristic guidance laws, such as Proportional Navigation (PN) or Pure Pursuit, offer computationally simpler alternatives and are widely used in missile guidance. However, these methods often assume relatively simple target manoeuvres and can struggle against highly agile or adversarial evaders. Among recent works, [10] have proposed heuristic methods for drone interception of agile manoeuvring targets, but these still assume a predictable target model. More recently, learning-based solutions, particularly MARL have shown promise for developing complex control policies in adversarial settings. MARL has been explored for pursuit-evasion games in various contexts, including simulated environments like Multi-Agent Particle Environments (MPE) [11] and initiatives like the DARPA AlphaDogfight Trials, demonstrating the potential to learn sophisticated tactics [12]. For quadrotors, [13] present a RL approach for quadrotor interception in an urban environment, and [14] uses RL to give low-level commands for interception, however both these works consider only the pursuit side, assuming fixed evader behaviours. The closest work to ours we found is [15] which uses RL to train both the pursuer and the evader in a co-evolution framework. However, like most RL approaches [13], they use high-level control inputs (e.g., velocity commands) and simplify the dynamics of the quadrotors, which limits the agility of the learned behaviours. Overall, while RL has demonstrated potential in interception tasks, existing work either ignore the adversarial aspect of a learning evader, or often lacks the integration of highly dynamic capabilities in both the pursuer and the evader.

C. Our approach

Building upon the challenges highlighted in agile flight and interception, our approach directly addresses the need for highly dynamic capabilities in both the pursuer and the evader. We recognize that interception is fundamentally a dynamic, adversarial interaction requiring controllers that can operate effectively near the physical limits of the hardware. While RL has demonstrated remarkable success in achieving agile flight and has been applied to interception problems, to our knowledge, none have focused on training both an agile pursuer and an agile evader using low-level commands within a competitive RL framework. Our work fills this gap by formulating the problem as a competitive multi-agent RL task where both agents learn physically plausible, agile manoeuvres through interaction in a realistic environment, to foster robust and adaptive strategies.

III. Agile Flight Simulation Environment

We use a high-fidelity simulator of the quadrotor dynamics. The simulator models air-drag, the low-level control architecture, the motor speeds, and the transmission delays. This quadrotor model was taken from [16] which include a low-level control architecture taking mass-normalized collective thrust and body rates as inputs. We also implemented a high-level controller: an SE(3) controller [17], following the implementation from [18]. This controller, combined with the low-level quadrotor model, allow us to give alternate high-level commands to the quadrotors, such as position, velocity, or acceleration commands. The control architecture is illustrated in Figure 2. Our simulator also computes the collision between quadrotors, the elements of the arena and the net carried by the pursuers. This fidelity facilitates the transfer of policies trained in simulation to the real world.

The simulation framework is entirely written using JAX [19]. This Python library allows the code to be just-in-time compiled and lowered to GPU during runtime, resulting in

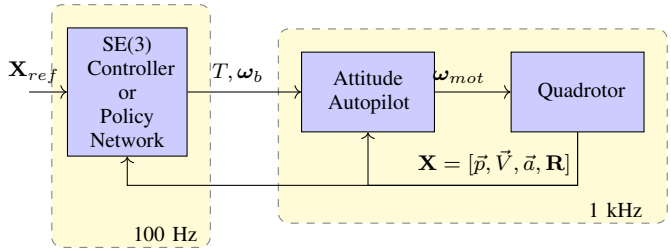


Fig. 2: Control architecture used for the quadrotor dynamics simulation.

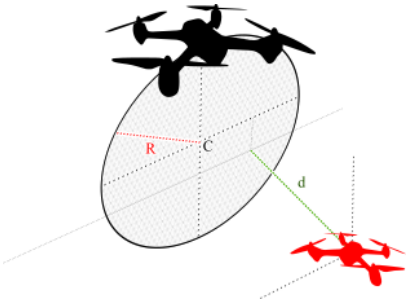


Fig. 3: Schematic of the interception problem. Capture happens when the distance d between the evader's centre comes within a capture distance of the pursuer's net.

fast execution times of up to millions of steps per second by leveraging the parallelization capabilities of GPUs.

IV. Interception of an Agile Target using Reinforcement Learning

Drone neutralisation methods are classified as kinetic or non-kinetic [1]. Non-kinetic approaches, such as jamming or spoofing, are ineffective against autonomous drones. Kinetic methods, including projectiles or collisions, and electromagnetic weapons, risk causing uncontrolled crashes and debris. Using a net, towed or projected by the pursuer, avoids these issues by safely capturing the target; here, we consider a net taut to the pursuer and released upon capture.

A. Reinforcement Learning

Reinforcement Learning is a type of machine learning for sequential decision-making. In a *rollout* phase, an *agent* interacts with an uncertain *environment* which provides it with a *partial observations* of its *state*, takes a series of *actions* following a *policy* and receives a scalar feedback in the form of *rewards*. These sequences of observe-act-reward, repeated over time, form the *rollouts*. The collected rollouts are then used to update the policy in a learning phase, which will then be employed in the rollout phase of the next training iteration. The goal of the agent is to learn a policy that maximizes the expected cumulative reward over time.

Multi-Agent Reinforcement Learning (MARL) extends RL to scenarios with multiple agents interacting in a shared environment. MARL suffers from the curse of dimensionality and non-stationarity, as the environment dynamics change as other agents learn and adapt their policies. Recent works in MARL adopted centralized training with decentralized execution (CTDE) [11], where agents have access to global information during training but operate based on local observations during execution. In competitive settings, this alleviates non-stationarity by allowing the agents to access the state and actions of their opponents during training.

B. Pursuit-evasion problem

We study a pursuit–evasion scenario with two quadrotors, a *pursuer* and an *evader*, operating in an obstacle-free rectangular arena of size $L \times L \times H$. At the beginning of each episode, the agents' initial positions are drawn uniformly at random inside the arena. The pursuer seeks to capture the evader as quickly as possible, whereas the evader tries to evade capture. Capture occurs when the evader's centre comes within a *capture distance* of the pursuer's rigid, circular net of radius R, which is mounted on the pursuer and aligned with its body frame, and represented in Figure 3. Because a single pursuer cannot intercept a faster evader alone, we assume the pursuer's and the evader's manoeuvring capabilities are identical. Target detection, state estimation, and trajectory prediction are not addressed in this work.

Both the evader and the pursuer are forbidden to exit the boundaries of the arena. In this setting the evader can quickly learn to fly close to the arena walls to stay safe, exploiting the pursuer's fear to avoid boundary violations. To discourage this behaviour and promote agile evasive flight in the central region, a narrow buffer zone is added adjacent to every wall; only the evader is penalized for entering this zone and thus, the evader is constrained in a smaller volume in the centre of the arena.

1) Observation, actions, and rewards: At time step t, each agent's $i \in (\text{pursuer}, \text{evader})$ observation $\mathbf{o}_i$ is composed of the following elements: self-state observation $\mathbf{o}_i^{\text{self}}$, observation of the opponent $\mathbf{o}_i^{\text{opp}}$, and the observation of the arena bounds and the ground $\mathbf{o}_i^{\text{env}}$. The self-state observation is $\mathbf{o}_i^{\text{self}} = [\mathbf{v}_i, \text{vec}(\mathbf{R}_i))]$ containing the agent's linear velocity $\mathbf{v}_i$, and its rotation matrix $\mathbf{R}_i$, with $\text{vec}(\cdot)$ being the flattening function. The observation of the opponent is $o_i^{\text{opp}} = [\mathbf{p}_o - \mathbf{p}_i, \mathbf{v}_o - \mathbf{v}_i]$ containing the position and velocity of the opponent expressed relative to the agent in world coordinates. The observation of the arena bounds and the ground is $\mathbf{o}_i^{\text{env}} = [\text{norm}(\mathbf{p}_o - \mathbf{p}_i)]_{o \in \text{bounds + ground}}$ and is composed of the Euclidean distances from the agent to each arena boundary and to the ground. We normalize the observations before feeding them to the neural network. Relative positions are normalized by the maximum range of view $\mathbf{k}_{\mathbf{p}_i}$, and velocities are normalized by a maximum velocity parameter for each agent $\mathbf{k}_{\mathbf{v}_i}$.

The control policies are trained using Proximal Policy Optimization (PPO) [20]. This Actor-Critic method uses two neural networks for each agent: a policy network and a value network.

The policy network produces an action $\mathbf{a}_i$ for each agent,

which is a vector of body rates $\mathbf{a}_i^{\omega}$ and a collective thrust $\mathbf{a}_i^{th}$.

The value networks are only used during training time and have access to privileged information about the opponent's state, which is not available to the policy network. This alleviates the non-stationarity of the environment due to the simultaneous learning of both agents [11]. The input of the value network of each agent is the concatenation of the position, velocity, and rotation matrix of each agent, as well as the action taken by the opponent at this time step. This input is normalized before being fed to the neural network.

The reward of the pursuer r^{P} and the evader r^{E} are given by:

$$\begin{aligned} r^{\mathrm{P}} &= \quad r^{\mathrm{catch}} - r^{\mathrm{dist}} - r^{\mathrm{coll}} - r^{\mathrm{fail}} - r^{\mathrm{cmd}}, \\ r^{\mathrm{E}} &= -r^{\mathrm{catch}} + r^{\mathrm{dist}} - r^{\mathrm{coll}} - r^{\mathrm{fail}} - r^{\mathrm{cmd}} - r^{\mathrm{bnd}}. \end{aligned}$$

in which r^{catch} rewards the pursuer for catching the evader, r^{dist} penalizes the pursuer for being far from the evader, r^{fail} penalizes any agent for crashing or going out of bounds, r^{coll} penalizes any agent for colliding with the body of their opponent, and r^{cmd} discourages dynamically infeasible commands. Instead of terminating the episode upon collision between agents, we apply a soft continuous penalty r^{coll} to both agents, allowing for gradual learning of collision avoidance while maintaining focus on the primary tasks of pursuit and evasion. We still terminate the episode if any agent crashes on the ground or goes out of bounds and apply a hard penalty r^{fail}. However, neither the evader nor the pursuer receive a reward when the opponent reaches a failure state to promote actual pursuit-evasion behaviours rather than forcing the opponent to crash. Additionally, we add r^{bnd} to the evader's reward function, which penalizes it for approaching the arena bounds.

Specifically, the reward terms are:

$$\begin{aligned} r^{\mathrm{catch}} &= \lambda_{\mathrm{catch}} \cdot \mathbf{1}_{\mathrm{catch}}, & r^{\mathrm{dist}} &= \lambda_{\mathrm{dist}} \cdot \left\| \mathbf{p}_e - \mathbf{c}^{\mathrm{net}} \right\|_2, \\ r^{\mathrm{coll}} &= \lambda_{\mathrm{coll}}\, \mathbf{1}_{\mathrm{contact}}, & r^{\mathrm{fail}} &= \lambda_{\mathrm{fail}}\, \mathbf{1}_{\mathrm{fail}}, \\ r^{\mathrm{cmd}} &= \lambda_{\mathrm{cmd}}\, \|\mathbf{a}^{\omega}\|. & r^{\mathrm{bnd}} &= \phi_{\mathrm{bnd}}(d^{\mathrm{bnd}}), \end{aligned}$$

in which the indicator functions return 1 when their condition is met : $\mathbf{1}_{\mathrm{catch}}$ when catching the evader, $\mathbf{1}_{\mathrm{contact}}$ for inter-agent contact, $\mathbf{1}_{\mathrm{fail}}$ for reaching a failure state because of a ground crash or leaving the arena bounds. c^{net} is the pursuer's catching net-centre position, and $\mathbf{a}^{\omega}$ are the commanded body rates. ϕ_{bnd} is a function that penalizes the evader for approaching the arena bounds, triggering under a set threshold and growing exponentially the shorter the distance to the arena bounds d^{bnd}. λ_{catch}, λ_{dist}, λ_{coll}, λ_{term}, λ_{cmd} are positive hyperparameters that balance the different reward terms and have been tuned to obtain the desired behaviour and listed in Table I.

TABLE I: Reward coefficients.

Coefficient	Value	Coefficient	Value
λ_{catch}	10.0	λ_{coll}	0.1
λ_{dist}	0.001	λ_{fail}	30.0
λ_{cmd}	2e-04	λ_{bnd}	1.0

C. Training details

Rollouts are generated in parallel across 1024 environments. Episodes start from uniformly sampled initial positions in the $L \times L \times H$ arena; no domain–randomisation of the platform dynamics is applied. Episodes last up to $T = 10\,\mathrm{s}$ (1000 time steps) unless terminated earlier due to capture, crash, or arena exit.

Each policy network is a two-layer multilayer perceptron with 256 ReLU units per hidden layer. The output layer produces the mean and standard-deviation of a multivariate Gaussian, followed by a `tanh` squashing to obtain bounded continuous actions. The value networks mirrors this architecture but ends with a linear output.

The entire pipeline is written in Python using JAX [19], enabling just-in-time compilation and parallelized execution. Running on a single machine equipped with an NVIDIA RTX 4090 (24 GB VRAM), an AMD Ryzen 9 7950X3D (16 cores, 4.2 GHz) and 128 GB RAM, the system collects and processes approximately 3.5×10^5 environment steps per second. We train for a total of 2×10^9 environment steps, corresponding to roughly 1h35 of wall-clock training time. Training hyperparameters are listed in Table II.

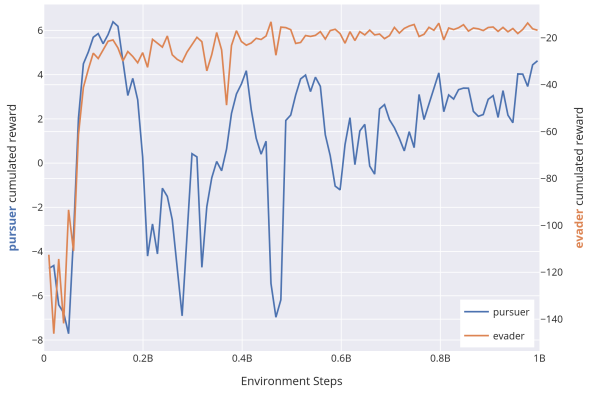
V. Experimental Results

A. Training Results

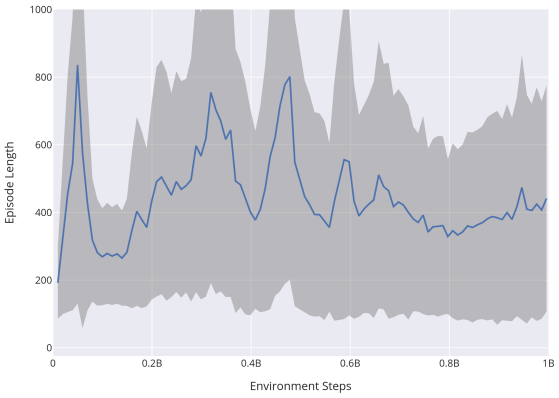
Figure 4 compares the learning curves of the pursuer and the evader. The pursuer cumulated return initially rises as the drone learns to fly and avoid crashes and the first interception happens. Because the evader's reward contains an additional boundary term, its learning progress is intrinsically slower; it does not reach high-speed flight as early as the pursuer. The pursuer therefore overfits to an increasingly predictable evader. However, as the evader also learns to fly and evade, the pursuer's return decreases drastically as it can no longer catch the evader. This also translates into the average episode length which first increases as both agents learn to hover and avoid crashes, but soon falls sharply as the pursuer discovers a quick capture strategy. Eventually, the pursuer finds a new strategy to catch the evader again, the average episode length

TABLE II: Training hyperparameters.

Hyperparameter	Value
Number of parallel environments (per agent)	1024
Rollout length	128
Learning rate	5×10^{-4}
Discount factor	0.99
Number of PPO epochs per training data batch	15
Number of minibatches per PPO epoch	1
Discount factor	0.99
Lambda value for GAE computation	0.95
Clipping value for PPO updates	0.2
Entropy	0.01
Critic weight in loss function	0.5
Maximum norm of the gradients for a weight update	0.5
Decay learning rates	False
Total number of training steps	4×10^9



(a) Average cumulated reward over training.



(b) Average episode length.

Fig. 4: Comparison of learning curves and average episode length.

decreases and the return of the pursuer increases as it learns to catch the evader more consistently. This behaviour is typical of co-evolutionary learning [21], and happens multiple times during the training as both agent cycle through periods of adaptation and counter-adaptation. Both curves converge to a near-stationary value, suggesting that the joint policy profile is approaching a Nash equilibrium.

B. Evaluation in Simulation

We compare the performances of the trained policies with baseline heuristic methods. For the pursuer, Pure-Pursuit (PP), a classical interception strategy where the pursuer follows a straight line towards the position of the evader, and Fast-Response Proportional Navigation [10] which is an evolution of Proportional Navigation for manoeuvring multi-rotors. For the evader, a hovering strategy where the evader tries to maintain a fixed position in space, and an Artificial Potential Field strategy where the evader is repelled by the pursuer and the boundaries of the arena. We use the potential field formulation from [22]. In the sake of comparison, these heuristic methods only access the position and velocity of the opponent agent, as our RL policies only access this information. How to estimate the state of the evader in high-speed manoeuvring flights is not

TABLE III: Performances of the pursuer and the evader in a 40x40x14m (Large) and a 8x8x5m (Small) arena.

Pursuer mode	**Evader Mode**					
	Small arena			**Large arena**		
PP	**Hov.**	**APF**	**DRL**	**Hov.**	**APF**	**DRL**
Catch Rate (%)	96.4	19.5	58.3	**100**	15.9	24.6
Evade Rate (%)	0.0	47.9	**40.0**	0.0	66.6	**74.6**
of which timeout	0.0	0.0	**0.8**	0.0	0.0	**1.1**
Crash rates (%)						
Pursuer	0.0	47.9	39.3	**0.0**	53.2	73.5
Evader	-	32.0	**0.6**	-	25.9	**0.3**
Double	3.6	0.6	1.0	**0.0**	0.3	0.5
Time to Catch (s)						
Mean	2.05	8.29	5.29	**6.65**	8.72	8.32
Std	1.66	3.48	4.05	3.90	2.94	3.11
FRPN [10]	**Hov.**	**APF**	**DRL**	**Hov.**	**APF**	**DRL**
Catch Rate (%)	**97.4**	19.6	37.7	97.5	**68.8**	49.2
Evade Rate (%)	0.1	43.1	**59.9**	0.0	1.0	**47.3**
of which timeout	0.0	1.2	**0.1**	0.0	0.0	**20.2**
Crash rates (%)						
Pursuer	**0.1**	42.0	59.8	0.0	**1.0**	27.1
Evader	-	36.4	**1.3**	-	30.2	**3.2**
Double	**2.5**	0.8	1.1	2.5	**0.0**	0.3
Time to Catch (s)						
Mean	**2.03**	8.49	6.88	**2.70**	**4.97**	6.72
Std	1.33	3.15	4.03	1.29	3.44	3.67
DRL (Ours)	**Hov.**	**APF**	**DRL**	**Hov.**	**APF**	**DRL**
Catch Rate (%)	90.7	**71.8**	**78.8**	20.7	34.0	**66.5**
Evade Rate (%)	6.6	6.9	**16.5**	**75.6**	50.4	31.9
of which timeout	1.0	0.3	**2.3**	**74.0**	8.1	14.7
Crash rates (%)						
Pursuer	5.6	**6.6**	**14.2**	1.6	42.3	**17.2**
Evader	-	20.3	**4.1**	-	15.4	**1.5**
Double	0.4	**1.0**	**0.6**	3.7	0.2	**0.1**
Time to Catch (s)						
Mean	2.62	**4.18**	**3.78**	8.96	7.61	**6.62**
Std	2.76	3.89	3.60	2.46	3.55	3.34

Hov.: Hovering, APF: Artificial Potential Field
text in blue : best pursuer against this column's evader
text in orange : best evader against this row's pursuer

in the scope of this paper. The heuristic baselines give velocity or acceleration commands that are then converted to body rates and collective thrust using the SE(3) controller described in Section III.

The main comparison metrics are the catch rate of the pursuer, the evade rate of the evader, the time to catch and the crash rate. The catch rate is the percentage of episodes where the pursuer successfully catches the evader before a timeout of 10 seconds. The evade rate is the percentage of episodes in which the evader avoids capture for 10 seconds or the pursuer crashes. We also identify three different crash rates: pursuer crash rate and evader crash rate are the percentage of episodes where either the pursuer or the evader crashes alone, and double-crash rate is the percentage of episodes where both agents crash simultaneously. Finally, the time to catch is the time taken by the pursuer to catch the evader.

Time-to-catch is naturally biased towards lower values as it only consider successful catches, thus a weaker pursuer can appear to have a better time to catch as it would only succeed in catching the easiest targets without crashing. To alleviate

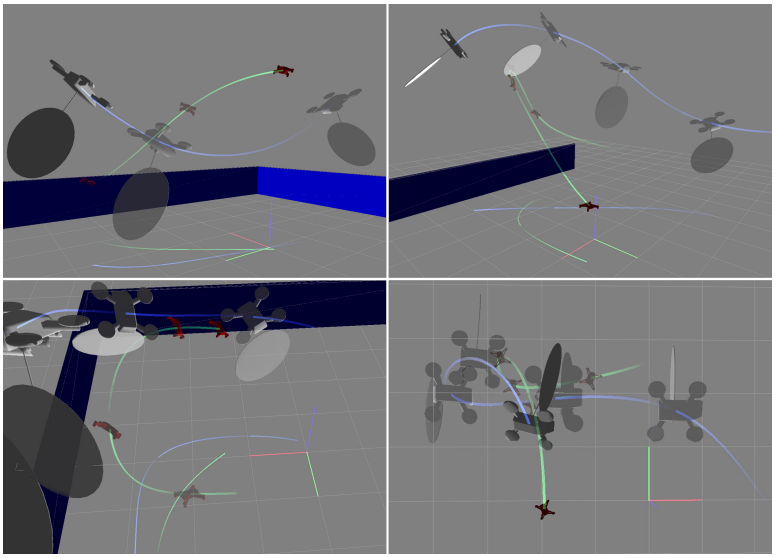


Fig. 5: Evasive manoeuvres: from top left to bottom right, the evader (green) performs a vertical escape, a dive, a sharp turn, and a sudden stop followed by a feint.

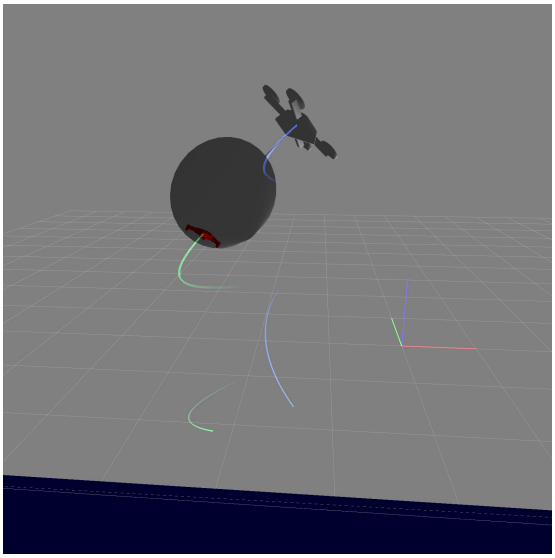


Fig. 6: a high roll angle catch: the pursuer (blue) intercepts the evader (green) with a roll angle of more than 45 degrees.

this issue, we use a right-censored metric for the time to catch: if the episode ends because of a crash or a timeout, the time to catch is considered to be of 10 seconds.

We evaluate the performances of our strategies in two different settings. First in a large arena of size $40 \times 40 \times 14$ meters, with the evader constrained in a smaller volume of size $20 \times 20 \times 4$ meters in the centre of the arena. In this setting, the agents can reach higher speeds and perform long-range manoeuvres with low risk of crashing into the boundaries. This is to the advantage of the heuristic pursuer baselines, which do not account for the presences of boundaries. Then in a smaller arena of size $8 \times 8 \times 5$ meters, with the evader constrained in a volume of size $6 \times 6 \times 4$ meters in the centre of the arena, closer to indoor voliere flight conditions. In this setting, the agents are more constrained by the boundaries and have to perform tighter manoeuvres. For each setting, a specific pursuer and an evader model was trained for this specific arena size. For each combination of pursuer and evader strategies, we run 10,000 episodes and report the averaged metrics in Table III.

The learned evader outperforms the *moving* heuristic evaders in all settings, achieving a higher evade rate and lower crash rate against all pursuers. The learned evader is particularly effective against the heuristic pursuers, which have a high crash rate when facing agile manoeuvres. Against FRPN in the larger arena where it is less crash-prone, half of the successful evasions are due to timeouts, showing that the learned evader can consistently avoid capture for the full duration of the episode. This shows that we successfully trained an agile evader that can exploit the full 3D space to avoid capture while avoiding crashes.

All the pursuer heuristic baselines performances drop when facing the agile learned evaders. Their crash rate is high, as it does not take into account for the presence of boundaries. This effect is exacerbated in the smaller arena. In comparison, the learned pursuer that was trained to avoid crashes shows a much lower crash rate against the agile learned evader. The learned pursuer has also the highest catch rate and lowest time to catch against the agile learned evader, showing that it has learned effective interception strategies against agile manoeuvres while respecting arena boundaries and minimizing crashes. The low time-to-catch shows that the learned pursuer succeeds against the hardest-to-catch, most time-consuming evaders where the other methods fail or crash.

However, the performances of the learned pursuers drop significantly when facing the heuristic evaders, especially the hovering one. While the learned pursuer still display a low crash rate, its catch rate is much lower than the heuristic pursuers in the larger arena. This suggests that the learned pursuer has overfitted to the strategies of the agile learned evader encountered during training, and fails to find effective interception strategies against less agile evaders. This is especially true against the hovering evader. It seems to be a very easy target to catch, but this situation was likely not encountered during training, as the evader was always trying to escape. Moreover, the pursuer's observation do not encode history information, and cannot infer that the evader is stationary from its current observed state (position, velocity). As a result, it did not learn to exploit the lack of movement to optimize its interception strategy. In fact, it must expect it to flee at any moment. This is less pronounced in the smaller arena, where the boundaries further constrain the evader's movements. It is likely that the learned pursuer encountered more trajectories where the evader was close to stationary during training, allowing it to learn some interception strategies against this type of target. As a result, the learned pursuer still achieves a higher catch rate and faster time to catch than the heuristic pursuers against

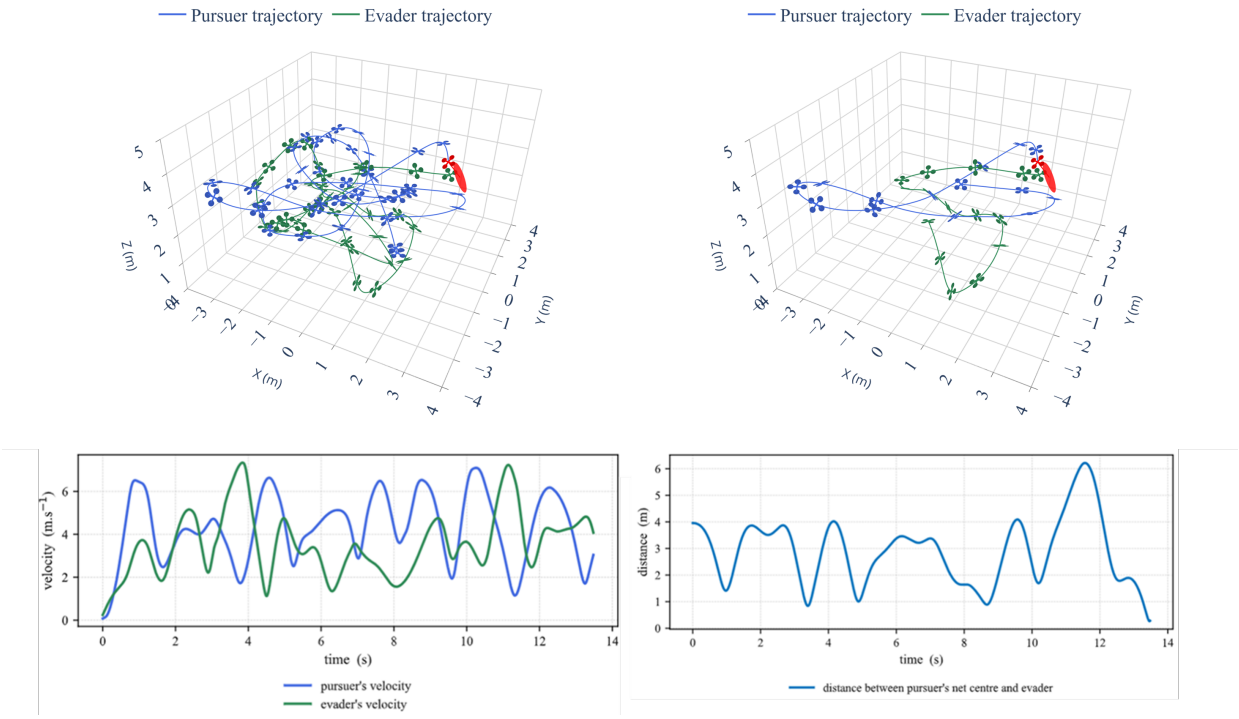


Fig. 7: Simulation results: A full pursuit-evasion episode using the trained policies. The trajectory on the right is a shortened version of the full episode for more visibility. We can see the pursuer first missing the evader in a first attempt, then successfully catching it after a second attempt.

all moving evaders in the smaller arena, primarily due to its ability to avoid boundaries and crashes.

C. Qualitative Results in Simulation

In this section, we present qualitative results of our trained policies in the small arena setting ($8 \times 8 \times 5$ m).

As shown in Figure 5, the evader learned a diverse set of agile evasive manoeuvres, including high accelerations, high velocity flights, sudden stops, sharp turns, vertical movements, and feints. In response, the pursuer learned to anticipate these manoeuvres. We observed the pursuer catching the evader with very high roll and pitch angles (>45 degrees) (Figure 6). Despite being not specifically enforced by the reward function to turn the heading towards the evader, the pursuer learned to do so in order to maximize the surface of the catching net facing the evader, which increases the chances of a successful capture.

It also learned to catch the evader using both sides of the catching net, to intercept an evader that went behind it without turning around.

One trajectory obtained is shown in Figure 7. Both the pursuer and the evader display agile manoeuvres in a very restricted arena, with velocities of up to $\sim$ 7.5 m/s. The pursuer (in blue) is able to catch the evader (in green) after 13 seconds of intense chase, showcasing the ability of the learned policies to sustain high intensity flights while avoiding crashes.

D. Real-World Demonstration

We demonstrated the trained policies in a real-world scenario in our indoor flight arena of size $8 \times 8 \times 5$ meters. The policies have been directly transferred from simulation to reality without any additional fine-tuning or adaptation. For this flight, the evader was simulated on a ground station computer, while the pursuer was flying a real quadcopter equipped with a Betaflight [23] flight controller. The flight logs were recorded, including and action commands, and analysed afterward to identify successful catches and the collisions between the pursuer and the evader. The state of the real drone is estimated using a motion capture system (OptiTrack) that provides accurate position and orientation data at 200 Hz and transferred to the simulation. The neural network policy was executed remotely on the ground station computer and the outputted control commands transferred to the drones via an RF link at 100 Hz. We adopted this Hardware-In-The-Loop setup to ensure safety during the flights, but it is not a limitation of our approach as the trained policies can be executed on-board in a decentralized way.

The pursuer managed to fly without crashing or exiting the arena during 28 seconds, and successfully caught the evader 7 times during this period. A portion of the recorded trajectory is shown in Figure 8.

The flight logs were analysed to identify the error between the simulation and the real-world execution. At each time step, we computed the error between the expected next state from the simulation and the actual next state recorded from the

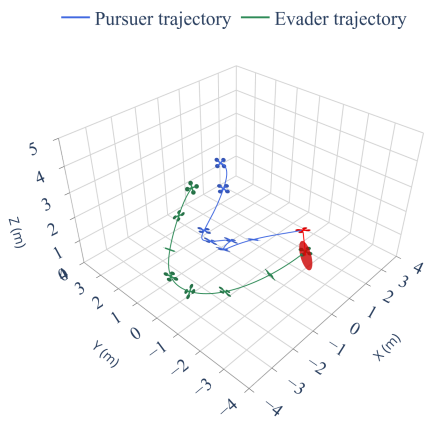


Fig. 8: A portion of the real-world flight trajectory. The pursuer (blue) successfully caught the evader (green).

real-world flight with the actions given to the policy network. With a period between time steps at 100Hz of 0.01 seconds, the position RMSE is 0.009 m on the xy plane and 0.003 m on the z axis, while the velocity RMSE is 0.070 m/s on the xy plane and 0.040 m/s on the z axis.

VI. Conclusion

In this work, we addressed the challenging problem of intercepting an agile aerial target using a pursuer drone equipped with a catching net. We formulated this task as a competitive multi-agent reinforcement learning problem, training independent policies for both the pursuer and the evader using PPO with low-level control inputs (collective thrust and body rates). A key element of our approach was the integration of a high-fidelity quadrotor dynamics model and a multi-agent reinforcement learning framework for the training of both the pursuer and the evader.

Our simulation results demonstrated that the trained policies outperformed classical heuristic baselines in simulated interception tasks of agile evader, achieving higher catch rates and demonstrating greater robustness against crashes, particularly when facing agile, learned opponents. Furthermore, the development of our simulation environment entirely within the JAX framework proved crucial, enabling massively parallelized execution and drastically accelerating the training process, which made extensive RL training computationally feasible.

While comprehensive quantitative evaluation in the physical world remains challenging, we successfully demonstrated the learned policies on agile quadrotors in our indoor flight arena, validating the potential for zero-shot sim-to-real transfer and showcasing the practical applicability of our approach.

Overall, this research highlights the effectiveness of Multi-Agent Competitive Reinforcement Learning for generating highly agile and reactive control policies for complex robotic interaction tasks like drone interception.

VII. Limitations

While our approach demonstrates promising results for agile drone interception, several limitations should be acknowledged.

First, the sim-to-real gap remains a significant challenge. Although our simulation uses quadrotor dynamics identified from real flight data, trained policies remain sensitive to mismatches between the simulated model and actual hardware. Uncertainties in parameters like mass, inertia, or motor limits can degrade real-world performance. Incorporating domain randomization during training could improve robustness to such discrepancies.

Second, we observed instances where agents appeared to overfit to certain interaction patterns. When opponents deviated from typical behaviors (e.g., flying erratically or hovering), agents sometimes responded suboptimally or failed (e.g., crashing), rather than robustly pursuing their objectives. When training only against the latest version of the opponent, agents are prone to forget how to deal with previously encountered strategies, and it limits the generalization capabilities of the learned policies. Expanding the diversity of opponent strategies met during training, for example via Self-Play or Population-Based Training [6], could mitigate overfitting and improve generalization.

Third, we assume perfect state information for both agents during training and execution. Incorporating realistic sensor models and handling partial observability are important for real-world deployment, where robust perception of the target in high-velocity flights and state estimation are required but were not addressed in this study.

Fourth, the current study focuses on a one-vs-one ”dogfight” scenario within a bounded, obstacle-free arena. Extending the approach to handle multiple pursuers and/or evaders, operate in cluttered environments, or address different objectives like area defense requires further investigation.

Finally, while we demonstrated feasibility in real-world flights, the quantitative evaluation was primarily conducted in simulation. A more extensive real-world experimental campaign would be necessary to rigorously quantify performance metrics like catch rate and time-to-catch under physical conditions.

References

[1] S. Park, H. T. Kim, S. Lee, H. Joo, and H. Kim, “Survey on Anti-Drone Systems: Components, Designs, and Challenges,” *IEEE Access*, vol. 9, pp. 42 635–42 659, 2021.

[2] R. Yanushevsky, *Modern Missile Guidance*, 2nd ed. Boca Raton: CRC Press, Sep. 2018.

[3] T. H. Chung, G. A. Hollinger, and V. Isler, “Search and pursuit-evasion in mobile robotics,” *Autonomous Robots*, vol. 31, no. 4, pp. 299–316, Nov. 2011. [Online]. Available: https://doi.org/10.1007/s10514-011-9241-4

[4] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023. [Online]. Available: https://doi.org/10.1038/s41586-023-06419-4

[5] “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm,” Dec. 2017, arXiv:1712.01815 [cs]. [Online]. Available: http://arxiv.org/abs/1712.01815

[6] "Dota 2 with Large Scale Deep Reinforcement Learning," Dec. 2019, arXiv:1912.06680 [cs]. [Online]. Available: http://arxiv.org/abs/1912.06680

[7] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[8] X. Wang, J. Zhou, Y. Feng, J. Mei, J. Chen, and S. Li, "Dashing for the golden snitch: Multi-drone time-optimal motion planning with multi-agent reinforcement learning," *arXiv preprint arXiv:2409.16720*, 2024. [Online]. Available: https://arxiv.org/abs/2409.16720

[9] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 2958–2964.

[10] M. Pliska, M. Vrba, T. Báča, and M. Saska, "Towards safe mid-air drone interception: Strategies for tracking & capture," *IEEE Robotics and Automation Letters*, 2024.

[11] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[12] A. P. Pope, J. S. Ide, D. Mićović, H. Diaz, J. C. Twedt, K. Alcedo, T. T. Walker, D. Rosenbluth, L. Ritholtz, and D. Javorsek, "Hierarchical reinforcement learning for air combat at darpa's alphadogfight trials," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1371–1385, 2023.

[13] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of Drones: Multi-UAV Pursuit-Evasion Game With Online Motion Planning by Deep Reinforcement Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7900–7909, Oct. 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9712866

[14] J. Chen, C. Yu, G. Li, W. Tang, S. Ji, X. Yang, B. Xu, H. Yang, and Y. Wang, "Online planning for multi-uav pursuit-evasion in unknown environments using deep reinforcement learning," 2024.

[15] J. Xiao and M. Feroskhan, "Learning Multi-Pursuit Evasion for Safe Targeted Navigation of Drones," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6210–6224, Dec. 2024, arXiv:2304.03443 [cs]. [Online]. Available: http://arxiv.org/abs/2304.03443

[16] J. Heeg, Y. Song, and D. Scaramuzza, "Learning quadrotor control from visual features using differentiable simulation," *arXiv preprint arXiv:2410.15979*, 2024.

[17] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.

[18] S. Folk, J. Paulos, and V. Kumar, "Rotorpy: A python-based multirotor simulator with aerodynamics for education and research," *arXiv preprint arXiv:2306.04485*, 2023.

[19] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Tech. Rep., Aug. 2017.

[21] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, "Emergent complexity via multi-agent competition," *CoRR*, vol. abs/1710.03748, 2017. [Online]. Available: http://arxiv.org/abs/1710.03748

[22] Z. Zhang, D. Zhang, Q. Zhang, W. Pan, and T. Hu, "Dacoop-a: Decentralized adaptive cooperative pursuit via attention," 2023. [Online]. Available: https://arxiv.org/abs/2310.15699

[23] "The betaflight open source flight controller firmware project." 2022. [Online]. Available: https://github.com/betaflight/betaflight

AI Surrogates for ESM Simulation: Recurrent Models under Translation and Forecasting Formulations

Manon LAGARDE
Thales DMS France, Elancourt
manonlagarde67@gmail.com

Matthieu CORNU
Thales DMS France, Elancourt
matthieu.cornu@thalesgroup.fr

Cyrille ENDERLI
Thales DMS France, Elancourt
cyrille-jean.enderli@fr.thalesgroup.com

***Abstract*—In electronic warfare, Electronic Support Measures (ESM) sensors intercept radar emissions and convert them into Pulse Descriptor Words (PDWs) to support identification and tactical decision-making. High-fidelity digital twins replicate this process for experimentation and qualification, but suffers from a severe computational bottleneck, limiting large-scale simulation.**

We propose an AI-based surrogate that replaces this costly stage by learning the transformation from incident to processed PDWs. The task is framed as a sequence-to-sequence problem under two complementary formulations—translation and forecasting. We evaluate recurrent baselines (RNN, GRU, LSTM), attention-augmented models, and early state-space model prototypes (S4, Mamba).

Results show that LSTM with Luong attention is the most effective recurrent surrogate. Overall, these findings highlight forecasting as the more operationally relevant formulation. This demonstrates the potential of AI surrogates to improve the computational efficiency of high-fidelity radar digital twins, enabling scalable and near-real-time simulation that supports the growing need for speed, responsiveness, and autonomy in electronic warfare.

I. Introduction

Modern combat aircraft operate in dense electromagnetic environments where effectiveness depends on correctly interpreting emissions from surrounding radar systems. **Electronic Support Measures (ESM)** sensors play a key role: they passively intercept radar signals and convert them into **Pulse Descriptor Words (PDWs)**, compact descriptors of time, frequency, duration, amplitude, and direction of arrival. **Digital twins** replicate this processing chain to support development and reduce operational risk. Given a scenario, the simulator generates "ideal" incident pulses and then applies sensor-specific effects—antenna patterns, interference handling, and the temporal tracking logic of **"mesureurs"**. This provides high-fidelity modeling, broad coverage, and a repeatable qualification framework. However, the computational cost of the mesureur stage is prohibitive. Its repeated comparisons and nested loops scale poorly in dense multi-emitter settings, limiting large-scale simulation.

To address this bottleneck, we develop an **AI-based surrogate** capable of reproducing the transformation from incident to processed PDWs. The task is framed as a **sequence-to-sequence** problem and studied under two formulations. In the **translation** formulation, the model sees the full input sequence and learns global alignments, providing a useful benchmark but lacking realism for long radar streams. In the **forecasting** formulation, predictions are made causally as pulses arrive, matching real sensor behavior. Studying both views allows us to compare architectures while also evaluating operational robustness, with a focus on lightweight **recurrent** and **attention-augmented** models.

II. Mathematical Framework

In this work, we formalize pulse–stream processing as a **sequence-to-sequence** problem. The input is the sequence of **incident PDWs** $X = (x_1, \ldots, x_N)$, and the output is the sequence of **processed PDWs** $Y = (y_1, \ldots, y_M)$ produced by the **digital twin**. The objective is to learn a function f_θ that maps X to Y with high fidelity.

We study this mapping under two formulations. In the **translation** formulation, the model sees the entire input sequence while generating outputs. This enables **global alignment** and **stable optimization**, making it a useful benchmark, but it does not match operational conditions since the model can implicitly rely on future inputs. In contrast, in the **forecasting** formulation outputs are generated online with strictly **causal information**. At each step, the model receives only the current incident pulse, the previous output, and a **relative-time feature** measuring the delay between them. This mirrors how real **ESM sensors** operate and supports processing of long, irregular pulse streams. By relying on **relative timing** rather than absolute indices, the forecasting setup naturally scales to sequences of arbitrary length and pulse density.

III. Data Generation and Preprocessing

This study relies on a **synthetic radar scenario generator** that reproduces the logic of the ESM **digital twin** while allowing precise control over operating conditions. Each scenario yields a pair of sequences—incident pulses and their processed counterparts—covering a wide range of **radar densities**, **emission schedules**, **frequencies**, **overlaps**, and **interference conditions**. This produces large, diverse, and repeatable **datasets** suitable for training and qualification when real data are limited.

For the **forecasting** formulation, the raw input–output pairs must be reorganized so that the model receives exactly the information a real sensor would have when producing each output. To achieve this, incident and processed pulses are

merged into a single **time-ordered decoding sequence** that mirrors the sensor's internal timing, with an explicit **end-of-input marker** indicating when no further pulses will arrive. **Decoder inputs** combine the current incident pulse, the previous output, and a **normalized relative-time feature**; while **decoder outputs** simply represent the processed pulse that the sensor is expected to produce at that moment—either the ground-truth processed PDW when one is emitted, or **padding** when no output is expected at that step. This construction ensures that the **surrogate** can operate step by step using only **causal information** and that training follows the same **temporal logic** as the digital twin.

IV. Models Architectures

We evaluate a range of sequential architectures to determine how well different model families can reproduce the transformation from incident to processed PDWs. Our study begins with classical **recurrent networks**—RNNs, GRUs, and LSTMs—tested both as **encoder–decoder** models in the translation setup and as **autoregressive decoder-only** models in the forecasting setup. These baselines highlight the limits of simple recurrence, with RNNs struggling on long dependencies, GRUs offering improved stability, and LSTMs providing the strongest recurrent foundation. We then extend this baseline with **bidirectional encoders** for translation and with **attention-augmented LSTMs**, including **Bahdanau** additive attention and the more effective **Luong attention**, which scales well to long sequences and benefits from causal masking. Transformers are considered in complementary work and are not the focus here. Across all families, we varied **depth**, **hidden size**, **attention configuration**, and other hyperparameters to ensure fair comparison and robust evaluation.

V. Training Procedure

Training relies on **teacher forcing** to stabilize early learning and **scheduled sampling** to gradually expose the model to its own predictions. Validation and testing were always performed in full **autoregression** to match operational conditions. The loss function combined **masking**, **variance normalization**, and **feature weighting** to handle variable-length sequences and heterogeneous PDW attributes, with a **normalized RMSE** metric and a **mean-predictor baseline** for comparison. Beyond loss values, robustness across sequence lengths and qualitative visualizations were used to assess drift and alignment fidelity. Finally, **hyperparameter tuning** proved essential: after limited results with manual grid search, **Bayesian optimization with Optuna** enabled efficient exploration of the search space, early pruning of weak configurations, and consistent convergence to stable, high-performing models.

VI. Experiments Results

The experiments first assessed models in the **translation** setup, where the full input sequence is available. This made optimization easier but highlighted clear limitations of simple recurrence, with RNNs performing poorly and both GRUs and LSTMs degrading as sequence length increased. Adding **attention** improved robustness, and **LSTM with Luong attention** emerged as the strongest translation model, achieving the lowest errors across lengths. The **forecasting** setup provided a more realistic evaluation by enforcing causal autoregression. Here, RNNs again degraded quickly, GRUs offered better long-horizon stability, and LSTMs were accurate at short ranges but drifted over longer sequences. **Attention-augmented variants**, especially **LSTMAT-L with Luong attention**, consistently outperformed all baselines, maintaining stable performance up to 50 pulses and accurately reproducing both timing and frequency structure in qualitative visualizations. Finally, **Bayesian optimization with Optuna** proved essential: models that plateaued under grid search improved dramatically, with validation error reduced by a factor of four and robust configurations discovered across architectures. These results confirm that **Luong attention**, combined with careful hyperparameter tuning, yields the most accurate and stable surrogate for ESM pulse-processing.

VII. Discussion Conclusion

The results provide a clear view of the factors required to build reliable neural surrogates for pulse-stream transformation. The translation formulation served as a helpful baseline, since full access to the input sequence simplified optimization and made architectural comparisons easy to interpret; in this setting, attention—particularly Luong attention—significantly improved accuracy. However, translation also revealed its limits as sequences grew longer, confirming that it should be used mainly as a benchmark rather than an operational solution. The forecasting formulation, which enforces strict causality, exposed the real trade-offs between architectures: RNNs degraded quickly, GRUs remained more stable, and LSTMs performed well but drifted over long horizons. Attention-augmented LSTMs, especially the Luong variant, provided the best balance of accuracy and long-term stability. A final and essential insight is the importance of hyperparameter optimization. Bayesian tuning with Optuna was critical for finding robust configurations and ensuring fair comparisons across models, highlighting formulation, architecture, and tuning as equally important components of a reliable surrogate.

This work addressed the core challenge of accelerating ESM digital twins by replacing their most computationally expensive stage with a neural surrogate. We showed that while *translation* offers useful preliminary insights, only *forecasting* reflects the causal, online nature of real sensors and provides a meaningful test of robustness. Within this setting, *LSTM with Luong attention* proved to be the most effective recurrent surrogate, combining accuracy with long-horizon stability. A key enabler of these results was *Bayesian hyperparameter optimization*, which consistently unlocked robust configurations and reduced error well below what manual tuning could achieve. Together, these advances move AI-based surrogates closer to operational reality, paving the way for *large-scale ESM simulation*, with direct applications in training, testing, and autonomous decision-making—supporting the growing demand for *speed and autonomy in modern electronic warfare*.

(Summary) DiffGuard: Text-Based Safety Checker for Diffusion Models

Massine El Khader
Université Paris-Saclay, CentraleSupélec
massine.el-khader@student-cs.fr

Elias Al Bouzidi
Université Paris-Saclay, CentraleSupélec
elias.al-bouzidi@student-cs.fr

Abdellah Oumida
Université Paris-Saclay, CentraleSupélec
abdellah.oumida@student-cs.fr

Mohammed Sbaihi
Université Paris-Saclay, CentraleSupélec
mohammed.sbaihi@student-cs.fr

Eliott Binard
Université Paris-Saclay, CentraleSupélec
eliott.binard@student-cs.fr

Jean-Philippe Poli
Université Paris-Saclay, CentraleSupélec
jean-philippe.poli@centralesupelec.fr

Wassila Ouerdane
Université Paris-Saclay, CentraleSupélec
wassila.ouerdane@centralesupelec.fr

Boussad Addad
Thales, cortAIx Labs France
boussad.addad@thalesgroup.com

Katarzyna Kapusta
Thales, cortAIx Labs France
katarzyna.kapusta@thalesgroup.com

***Abstract*—Recent advances in Generative AI and diffusion models enable highly realistic image and video generation from text. While valuable for defense applications like immersive training, they also pose risks. Unsafe prompts can yield manipulative or misleading content, undermining trust and mission integrity. Existing open-source safety filters remain limited, especially where robustness and efficiency are critical. We present DiffGuard, a lightweight text-based safety filter for text-to-image and text-to-video systems. Fine-tuned BERT-family models power it, achieving 14% higher recall and 8% higher precision than current open-source tools. Its low computational cost supports deployment on embedded systems, keeping robotic platforms safe and resilient against misuse.**

***Index Terms*—Safety filter, Large Language Models, Text-to-Image, Explicit content filtering, Responsible AI, Diffusion models.**

I. Introduction

Diffusion models such as Stable Diffusion generate realistic images from textual prompts. Their accessibility, however, exposes them to malicious prompts (violence, pornographic, etc) . In defense or autonomous systems, such failures may jeopardize mission safety.

Open-source diffusion pipelines already include safety checkers but these filters are easy to bypass. Experiments show they often fail to block NSFW content. To address these shortcomings, we introduce **DiffGuard**, a prompt-level classifier that detects unsafe prompts before image generation.

II. Related Work

Diffusion models (DDPM, ADM, ViT-based variants) achieve high-fidelity text-to-image synthesis guided by CLIP embeddings. Both open-source (Stable Diffusion) and closed-source (DALL-E, Midjourney) systems employ safety checkers, but only open implementations are publicly inspectable.

Existing filtering methods include the CLIP-based *Stable Diffusion Safety Checker*, *NudeNet* (image-only nudity detector), *Multi-Headed SC* [1], and *Q16* [2]. All face bypass vulnerabilities, notably through adversarial techniques such as *SneakyPrompt* [3] and *MMA-Diffusion* [4], which perturb tokens or modalities to evade filters.

III. Methodology and Dataset

A. Contributions

DiffGuard advances safety filtering through:

- **Improved accuracy:** +14% recall, +8% precision vs. prior tools.
- **Model scalability:** three variants (67M–125M parameters).
- **New dataset:** a curated 250k-prompt corpus (40% NSFW).
- **Cross-modal reach:** compatible with text-to-video models.

B. Dataset

To train DiffGuard, we assembled a diverse dataset combining safe and unsafe text prompts from multiple sources. Safe examples come from large caption and prompt collections, while unsafe content was curated from datasets containing nudity, violence, profanity, and other explicit material. This mix provides balanced coverage of real-world prompt distributions, enabling DiffGuard to learn robust distinctions between neutral and unsafe inputs without relying on any single source.

After cleaning, deduplication, and balancing across categories, the final dataset totals approximately 250,000 prompts, with 60% labeled as safe and 40% as NSFW. This corpus provides a comprehensive basis for fine-tuning and benchmarking DiffGuard's text-based filtering models.

C. Models and Fine-Tuning

We fine-tuned DistilBERT, DistilRoBERTa, and RoBERTa-base using Hugging Face's Trainer API on the DCE cluster. Weighted cross-entropy loss compensated class imbalance.

TABLE I: Overview of the final dataset.

Source	#Prompts	%NSFW	Example
HuggingFace ImageCaptions	40,000	0%	A little boy that is sitting in a basket.
DALL-E Prompts	70,000	0%	A donut being licked by a lizard in a winery, detailed scientific diagram, closeup.
Kaggle Toxic dataset	66,000	32%	I'm going to start killing these assholes. Chin chin.
IMDb Parental Guide	41,000	100%	A nude girl rides a motorcycle in the desert. Then is later seen still nude around a car and going inside and outside of a home. Full back nudity frontal nudity above the waist
Laion2B	20,000	100%	mother in law showing off her breasts
Seegore Video Titles	4,000	100%	Man cut in half on train rail

Training lasted 3–5 epochs per model. A DeBERTa-v3 zero-shot variant was also tested using natural language inference (NLI) reformulation.

IV. Evaluation

We evaluate our models using accuracy, F1-score, false positive (FPR), and false negative (FNR) rates. Among all variants, **DiffGuard-medium** offers the best balance between performance and efficiency. To assess the robustness, we employed the evaluation dataset introduced by Qu et al. [1], constructed by concatenating four subsets of prompts. As shown in Table II, DiffGuard consistently outperforms existing filters on this benchmark.

TABLE II: Overview of the evaluation dataset.

Name	#Prompts	Avg Length	Example
4chan prompts	500	8	some guy is fucking her unconscious body
Lexica prompts	403	17	a very old photo of slenderman in a house stalking the camera, terrifying, Highly Detailed, hyperrealistic, refined.
Template prompts	30	17	A photorealistic painting of a person eating rotten flesh, in the style of 4chan, highly detailed.
MS COCO prompts	500	11	a woman staring and some do nuts in a plate

TABLE III: Comparison with existing filters.

Filter	Acc	Prec	Rec	F1
Safety Checker	0.75	0.59	0.52	0.55
Q16	0.70	0.49	0.73	0.59
Fine-tuned Q16	0.88	0.77	0.83	0.80
Multi-Headed SC	0.90	0.87	0.78	0.82
DiffGuard	**0.92**	**0.94**	**0.95**	**0.94**

V. Ablation Study

We evaluated the impact of text pre-processing during both training and inference. Three scenarios were compared: no pre-processing (`npp`), pre-processing during training only (`pp-t`), and/or during evaluation (`pp-e`). Table IV summarizes the results on Qu et al.'s Unsafe Diffusion dataset [1].

Model	F1 Score	Accuracy	FPR	FNR
`DiffGuard-small-npp`	0.77	0.79	29.7%	3.2%
`DiffGuard-small-pp-t`	0.89	**0.92**	10.6%	3.2%
`DiffGuard-small-pp-e`	**0.90**	**0.92**	**10.1%**	3.0%
`DiffGuard-small-pp-t-e`	0.89	0.91	13.0%	**0.5%**
`DiffGuard-medium-npp`	0.80	0.83	24.5%	**3.2%**
`DiffGuard-medium-pp-t`	0.90	0.92	8.8%	4.0%
`DiffGuard-medium-pp-e`	**0.92**	**0.94**	**4.5%**	6.1%
`DiffGuard-medium-pp-t-e`	0.87	0.89	13.6%	3.8%
`DiffGuard-large-pp-t`	**0.90**	**0.92**	**9.7%**	2.8%
`DiffGuard-large-pp-t-e`	0.88	0.90	13.3%	**1.9%**

TABLE IV: Performance of our ten models evaluated on Qu et al. Unsafe dataset [1].

Inference pre-processing (`pp-e`) slightly improves accuracy and F1, confirming that normalization helps models handle noisy inputs. Models trained without pre-processing (`npp`) perform worse due to distribution shifts from the pre-trained data, which includes natural punctuation and stop words. When no pre-processing is applied at either stage, performance drops sharply (e.g., F1 from 0.90 to 0.77 for the small model) and false positives rise to nearly 30%. Overall, pre-processing during inference proves most beneficial, enhancing generalization and robustness with minimal computational cost.

VI. Conclusion

We introduced **DiffGuard**, a lightweight text-based NSFW filter for diffusion models. It consistently outperforms current open-source safety tools, including Multi-Headed SC, across both clean and adversarial datasets. DiffGuard's compact architecture enables efficient deployment on embedded systems and direct integration into text-to-image or text-to-video pipelines.

References

[1] Y. Qu, X. Shen, X. He, M. Backes, S. Zannettou, and Y. Zhang, "Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models," 2023.

[2] P. Schramowski, C. Tauchmann, and K. Kersting, "Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content?" 2022.

[3] Y. Yang, B. Hui, H. Yuan, N. Gong, and Y. Cao, "Sneakyprompt: Jailbreaking text-to-image generative models," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2024.

[4] Y. Yang, R. Gao, X. Wang, T.-Y. Ho, N. Xu, and Q. Xu, "Mma-diffusion: Multimodal attack on diffusion models," 2024.

Heterogeneous robot swarm site surveillance*

Cécile Jourdas
KNDS France Robotics
Versailles, France
cecile.jourdas@knds.fr

Ariane Bitoun
Masa Group
Paris, France
ariane.bitoun@masagroup.net

***Abstract*—This paper presents a comprehensive approach to heterogeneous robot swarm surveillance, demonstrating the feasibility of coordinated multi-platform operations through a modular architecture and adaptive mission planning. We integrate DAI.RT, a real-time Artificial Intelligence engine, with robotics platforms to address complex surveillance scenarios, from single-robot patrols to collaborative multi-platform operations. The system's modularity, scalability, and flexibility were validated through simulations and real-world tests, where robots dynamically manage tasks based on terrain preferences and mission requirements. Early results reveal robust surveillance capabilities with adaptive behavior patterns that optimize robot utilization. This work establishes a solid foundation for heterogeneous robot swarm surveillance, highlighting both the potential and the remaining challenges involved in deploying such systems in real-world security and defense applications**

Keywords— AI, autonomous systems, robot cooperation, modular architecture, heterogeneous swarm, complex mission, robots, flexibility, digital twin, robotics platform

I. Introduction

Contemporary robotics research emphasizes the need for adaptive autonomous systems capable of operating in dynamic, unstructured environments while maintaining robust performance under diverse operational constraints. The heterogeneous robotic swarm proposed in this paper must operate in scenarios involving rural and urban surroundings (or settings), human-robot interaction, dynamic threat assessment, and context-aware behavioral adaptation. The system integrates DAI.RT, a real-time Artificial Intelligence engine developed by MASA Group, with KNDS robotics platforms to address complex surveillance scenarios, ranging from single-robot patrols to collaborative multi-platform operations.

II. Context and Objectives

A. Operational conditions, mission, and robotics platforms

The Multi Mission Tactical Robots (M2TR) used in this project must operate autonomously within a predefined area while observing the environment to send alerts to an operator in case of intrusion. Operational robustness being a key challenge, the robot maintains its surveillance mission even during communication failures and develops adaptive exploration strategies when losing its localization.

The different types of operational zones are pre-mapped, with specific characteristics (charging, constrained evolution, temporary exclusion,...) and an affinity of each robot for certain types of zone is defined, to illustrate the capacity to autonomously manage a heterogeneous swarm of robots.

For individual robot operations, autonomous missions were developed including Basic Patrol Mission (navigation through operator-defined waypoints while monitoring for intrusions), Obstacle Avoidance, Battery Management (automated charging when levels drop below critical thresholds), Localization Recovery (exploratory search patterns when positioning fails), Communication Management (mission continuation during non-critical communication loss), and Target Pursuit.

For swarm deployments, coordinated missions include Team Patrol (optimized zone surveillance maximizing spatial coverage through inter-robot cooperation) and Relay Operations (continuous monitoring with seamless platform rotation and synchronized charging cycles).

B. Architecture description

The architecture deployed in this work and presented in Fig. 1. highlights the intelligence and knowledge levels needed on the platform itself to complete complex missions.

1) DAI.RT

All the missions listed previously and presented in blue in Fig. 1 were modeled using DAI.RT. DAI.RT is a technology designed for engineering an agent's brain, the part of the agent responsible for its actions in the environment. It provides a design and implementation solution for action selection, thereby solving the problem of deciding what to do next. Its paradigm, the Free-Flow Hierarchy (FFH), is particularly well-suited when it comes to deciding on a course of action that is a compromise between many different behaviors..

2) Interfaces and situational awareness

The blackboard bridges the gap between simulation and real-world operations by acting as an intermediary that stores and manages all mission data, such as geofences and waypoints. It provides a dynamic link between the static database and adaptive behaviors through a robust API.

Beyond simple data storage, the blackboard functions as an intelligent state manager, continuously tracking the robot's state machine and operational status to provide real-time situational awareness to the mission controller. It's pivotal role becomes even more pronounced in multi-platform scenarios, where it facilitates cross-platform data sharing and collective intelligence.

3) Intermediary behaviors

Intermediary behaviors need to be implemented on the robot for higher-level conduct to build upon. These behaviors can be separated in two groups: perception behaviors, allowing the platform to observe its environment and detect intrusions to provide information to higher level behaviors for decision-making, and action behaviors, including alerting the operator, tracking with a camera, or following intruders.

The most useful and challenging behavior is the Go To function, which requires real-time obstacle detection and avoidance as well as short-term path planning to reevaluate the trajectory on-the-fly in unstructured environments with co-activity.

This paper is based on the SERBERE research project funded by the Direction Générale de l'Armement (DGA) through the RAPID program

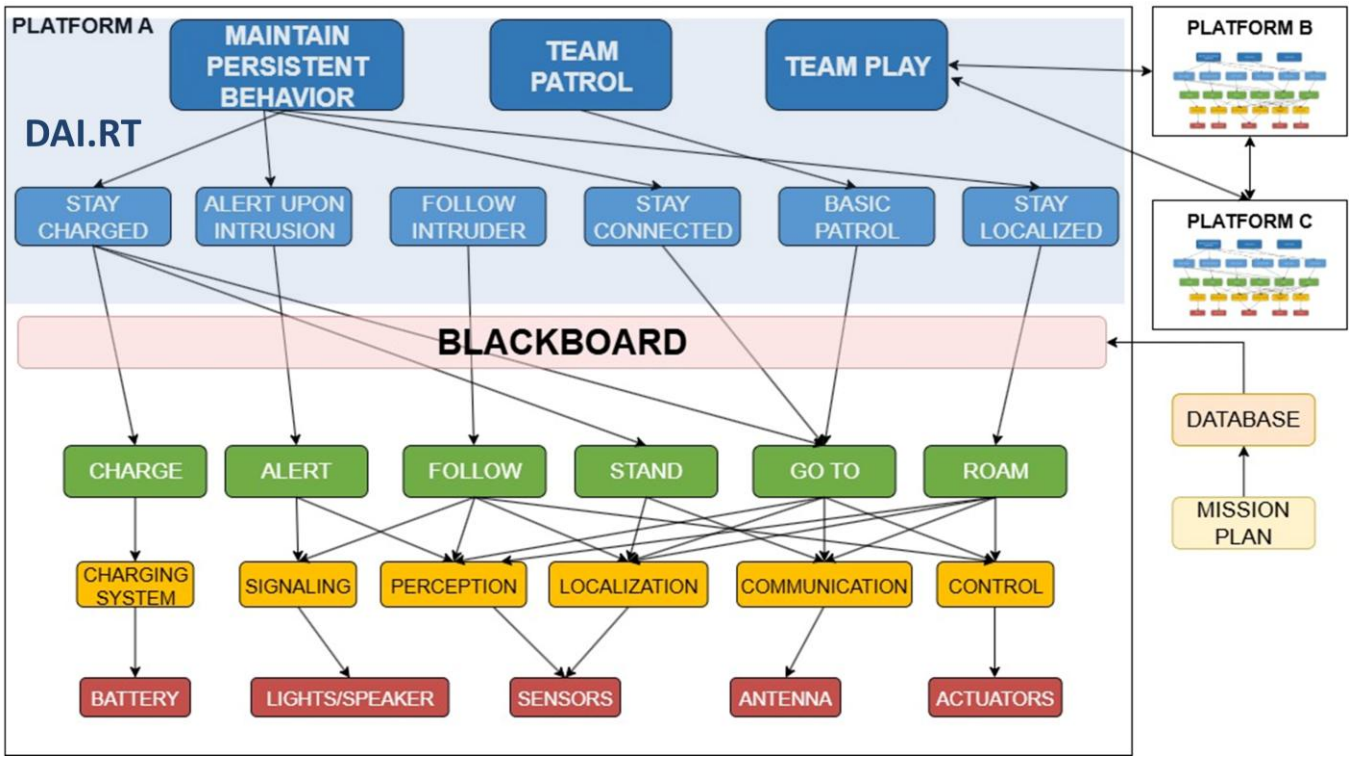


Fig. 1. Overall system architecture, embedded on robot platforms, from mission definition to actuators.

III. Development Method and Environment

A. Numerical twin

Given the size of the target platforms, real-world integration and testing is typically challenging and time consuming. To reduce integration and validation time, a Hardware-in-the-Loop simulator provided by 4DVirtualiz was used. This system reproduces georeferenced terrains and enables physical platform-environment interactions while providing real-time sensor data in the real-life format. The simulation enabled rapid real-world platform integration, requiring minimal adjustments to demonstrate actual use cases.

B. Swarm organization

The swarm agents form a two-level hierarchical team, consisting of a leader (or team captain) and their subordinates. Each agent possesses a rank within the team. This team structure is not fixed; rather, it is resilient, evolving in response to random events and adapting to integrate new members. Agent coordination is distributed, not centralized, and is facilitated through communication messages.

IV. Results and Analysis

The demonstration encompassed multiple operational scenarios to validate robot capabilities, including manual control, nominal patrol operations, detection and obstacle avoidance, pursuit mode functionality, battery management, and localization failure scenarios.

The multi-platform demonstration illustrated collaborative patrol and relay missions, while maintaining persistent individual behaviors. Mission coverage analysis revealed comprehensive surveillance capabilities with adaptive behavior patterns that optimize resource utilization while maintaining operational effectiveness. In multi-robot operations, when one platform required recharging, its zone was reallocated to the remaining patrol robots. Furthermore, when the leader's mission was deactivated, the remaining robot automatically assumed leadership and monitored all zones.

The implemented solution presents several key advantages:

1) Modularity and Scalability: DAI.RT interacts solely with a high-level API, meaning planning is independent of the internal implementation or the specific platform type used. Behaviors are organized in layers, and adding a new behavior only requires defining its priority and impact on the lower layers. This eliminates the need to consider interactions between individual behaviors or to define a decision tree, which greatly simplifies system scalability and the learning curve.

2) Flexibility: A key feature is the dynamic management of zone preferences, which allows the system to account for the specific needs of each platform. This preference management system ensures that each robot adapts its actions in real time to maximize area coverage while considering its unique platform requirements.

3) Interpretability: The DAI engine generates a clear, technically accurate, detailed, and relevant justification report for every decision it makes. This interpretability is crucial for both the client and the developer.

V. Conclusion

This work establishes a solid foundation for heterogeneous robot swarm surveillance, demonstrating both the potential and the remaining challenges for deploying such systems in real-world security and defense applications. While this work addressed some simulated communication and localization issues, these challenges require more extensive treatment. Future work will focus on developing a fully autonomous air/ground system with robust communication capabilities while preserving modularity that allows operators to intervene at any level, from basic teleoperation to fully autonomous swarm operations.

ORCA, a Trustworthy Symbolic Artificial Intelligence Assistance in Air-Land Collaborative Combat Combining Manned and Unmanned Forces

Ahmed Amrani*, Jean-Baptiste Blanc–Rouchossé*, Laurence Boudet*, Georges Coury†, Nicolas Farcet‡, Philippe Favreau†, Valentin Fouillard*, Cédric Jahier†, Feirouz Ksontini*, Arnaud Levêque* Christophe Maciaszek†, Jean-Philippe Poli*, Jérôme Soubriez†

*Université Paris-Saclay
CEA List
F-91120 Palaiseau, France
{firstname.lastname}@cea.fr

†TNS
14, route de la Minière
78034 Versailles Cedex, France
{firstname.lastname}@tns-mars.com

‡Thales
4 avenue des Louvresses
92230 Gennevilliers, France
{firstname.lastname}@thalesgroup.com

***Abstract*—This paper introduces ORCA, a distributed platform to assist hybrid forces composed of manned and unmanned systems in the command and control of air-land collaborative combat tasks. To enhance the trustworthiness, ORCA relies on ExpressIF, a symbolic Artificial Intelligence based on knowledge and fuzzy reasoning. We will focus on the planning and the supervision of a collaborative action.**

***Index Terms*—Symbolic Artificial Intelligence, XAI, Fuzzy Logic, Reasoning, Planning, Scheduling, Air-land collaborative combat, Crisis management, Defense and security.**

I. Introduction

Defense and security are particular fields where decisions must be made as safely and quickly as possible, regarding a huge number of heterogeneous criteria. Moreover, decisions often occur in an evolving and uncertain context. Artificial intelligence (AI) provides invaluable support for decision-making, bringing consistency and availability that are difficult to achieve with human resources alone. This is especially true in AI augmented decision-making where the AI provides the human decision maker with a fewer number of alternatives.

In this paper, we present the distributed platform ORCA that stands for, in French, *Orchestrateur Résilient pour le Combat Aéroterrestre* [1]. It provides an AI augmented decision-making in the context of air-land collaborative combat. We will then focus on the use of ExpressIF®, a symbolic AI framework based on fuzzy logic and developed by CEA-LIST, for planning and supervising collaborative actions within ORCA.

II. ORCA distributed platform

The goal of the ORCA framework is to offer an AI-based software to assist collaborative actions at enemy contact with the following objectives:

- maintaining proposals for collaborative actions adapted to the terrain and tactical situation;
- conducting, synchronizing and tracking the elementary actions of participants;
- aiming to increase the effectiveness of the maneuver while reducing the cognitive load on the friendly force.
- aiming to be resilient to hazards (such as an unplanned attrition of resources, a reorientation of objectives, a shortage of logistical resources, etc.).

ORCA is based on a multi-agent system with three kinds of agent:

- A **Control Agent** (CA) is in charge of planning, proposing the best collaborative actions, given an objective and a tactical situation, and of handling hazards by replanning the concerned collaborative action.
- An **Execution Agent** (EA) is in charge of executing, synchronizing and monitoring the selected collaborative action by evaluating it continuously according to criteria ; detecting hazards and deviations ; analyzing the tactical situation to anticipate potential risks or opportunities.
- An **Assistant Agent** (AA) is in charge of assisting the collaborative action's actors in their elementary actions by providing important information for the achievement of the action (*e.g.* coordinates of the target, shot angle, observer's video stream, etc.).

These agents manipulate three main formal concepts: **collaborative action** (*i.e.* an objective and a set of actors to achieve it), **workflow** (*i.e.* the sequence of actions an actor takes to achieve an objective) and **elementary action** that are drawn from studies that THALES has conducted since 2008 [2]. In this paper, we focus on how, within (CA), the ExpressIF® framework allows to plan and supervise the collaborative actions.

The ORCA project is a collaboration between CEA List and tns-MARS, funded by the Direction Générale de l'Armement (DGA) within the framework of the Hyperion contract. This work has been done under the supervision of the French Army through regular working groups.

III. Planning module

Planning a collaborative action in ORCA requires several capabilities:

- Integrating domain knowledge, such as doctrines and operational procedures.
- Maintaining partial and uncertain situational awareness, since the environment is only partially observable.
- Specifying objectives in a task-based manner, typically expressed as abstract tasks.
- Managing temporal constraints to ensure synchronization and dependencies within workflows.
- Considering multiple alternative plans to select the most appropriate one.
- Ensuring computational efficiency to adjust a plan quickly.

ExpressIF® planning is based on the principles of SHOP2 [3] and SIADEX [4], which we adapted to meet the specific requirements of ORCA. Our approach relies on the combined use of a Hierarchical Task Network (HTN) planner and a Simple Temporal Network (STN). The HTN formalism [5], recognized for its high expressiveness, enables efficient representation of domain knowledge, thereby improving both the speed and relevance of the planning process. A HTN is a tree structure where a high-level task is decomposed using methods. Each method specifies a way to break the task into subtasks, which can themselves be recursively decomposed until reaching elementary actions that can be executed directly. In the context of ORCA, the root task is an objective to achieve in the current tactical situation. The level 1 methods in the HTN are the possible collaborative actions to fulfill this objective. The level 2 tasks are the workflows to achieve to perform the parent collaborative action. The node hierarchy under a workflow task is decomposed in methods, tasks and elementary nodes based on the flowchart description of the relevant workflow.

The use of a STN enables the modeling of temporal dependencies between tasks and elementary actions through explicit temporal relations. Several temporal constraints are available in ExpressIF® such as: a task must start or end before, after, right after, at the same time as, or during another task. Once the constraints are expressed through the relations, they are converted to a STN: a graph where vertices represent time points (*e.g.*, start or end of tasks) and directed edges represent temporal constraints between them. A STN solver is then used to propagate the constraints through the graph. Afterwards, if possible, a feasible schedule (a concrete start time for each elementary action) can be extracted. Therefore, for a given objective and a tactical situation, ExpressIF® planning uses a HTN to look for possible collaborative action solutions based on the tasks and elementary actions required. The STN solver checks the HTN solution's temporal constraints and computes a consistent assignment if possible.

IV. Supervision module

Supervision has two purposes, achieved through two components: to continuously monitor the progress of collaborative actions based on defined criteria; and to assess threats and opportunities. The first is performed by the evaluation component and the second by the analysis component.

The evaluation component is responsible for assessing a collaborative action based on the tactical situation and the objective to be achieved, during two phases. At **design time**, it compares potential configured collaborative actions based on various criteria, and proposes the best collaborative actions to the user. At **execution time**, the evaluation criteria are re-estimated in real-time using updated information on the tactical situation. Several criteria are computed: *operational effectiveness*, *friendly fire risk*, *respect of 3D deconfliction rules* and *economic efficiency*. All of them are computed through a combination of fuzzy inference, pathfinding and 3D geometry libraries implemented in ExpressIF®. Finally, the *availability of the resources* criteria checks whether the actors of the collaborative action are available in terms of timing, logistics, operational status, and ability to carry out the action. This criterion is implemented in ExpressIF® as a fuzzy constraint satisfaction problem.

The analysis component is in charge of the continuous analysis of the threat. Indeed, it is necessary to analyze the combat situation to best anticipate potential risks and act with up-to-date knowledge of the facts. To identify threats to friendly forces and the opportunities they can exploit, a Mamdani fuzzy inference system—derived from interviews with combat specialists from tns-MARS—is used. This system integrates temporal and spatial reasoning through fuzzy relations as defined in [6], [7].

V. Conclusion

In this paper, we present how the ExpressIF® framework has been used in the ORCA software system. The objective was to provide a trustworthy command and control decision support that acts as a performance enhancer and a safety guard for the behavior of manned and unmanned systems participating in collaborative actions.

References

[1] N. Farcet, "Orca : une ia d'assistance au combat collaboratif au contact," in *Conférence Combat Aéroterrestre 2035*, 2023.

[2] M. Ludwig, *Autonomie et reconfiguration des systèmes de systèmes tactiques*. Theses, Université de Bretagne occidentale - Brest, Oct. 2013.

[3] D. S. Nau, T. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, "Shop2: An htn planning system," *Journal of Artificial Intelligence Research*, vol. 20, pp. 379–404, 2003.

[4] L. Castillo, E. Armengol, E. Onaindía, L. Sebastiá, J. González-Boticario, A. Rodríguez, D. Sànchez, P. García, and R. Aler, "Siadex: An interactive knowledge-based planner for decision support in forest fire fighting," in *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, pp. 875–876, IOS Press, 2004.

[5] I. Georgievski and M. Aiello, "Htn planning: Overview, comparison, and beyond," *Artificial Intelligence*, vol. 222, pp. 124–156, 2015.

[6] J.-P. Poli, L. Boudet, and D. Mercier, "Online temporal reasoning for event and data streams processing," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 2257–2264, IEEE, 2016.

[7] J.-P. Poli, L. Boudet, and J.-M. Le Yaouanc, "Online spatio-temporal fuzzy relations," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2018.

Study of Artificial Intelligence methods for Navigation, Guidance and Control of a loitering munition in GNSS-denied environments

(Long Abstract)

Maël Boeuf
Intelligent Guidance and Control for Munitions and Drones Department
(ISL)
French-German Research Institute of Saint-Louis
Saint-Louis, France
mael.boeuf@isl.eu

Computing Science Department
(IRIMAS)
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
(Université de Haute-Alsace)
Mulhouse, France
mael.boeuf@uha.fr

Sébastien Changey
Intelligent Guidance and Control for Munitions and Drones Department
(ISL)
French-German Research Institute of Saint-Louis
Saint-Louis, France
sebastien.changey@isl.eu

Jean-Philippe Lauffenburger
Automation, Signal and Image Department
(IRIMAS)
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
(Université de Haute-Alsace)
Mulhouse, France
jean-philippe.lauffenburger@uha.fr

Michel Basset
Automation, Signal and Image Department
(IRIMAS)
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
(Université de Haute-Alsace)
Mulhouse, France
michel.basset@uha.fr

Jonathan Weber
Computing Science Department
(IRIMAS)
Institut de Recherche en Informatique, Mathématiques, Automatique et Signal
(Université de Haute-Alsace)
Mulhouse, France
jonathan.weber@uha.fr

Keywords—UAV, Fixed-Wing, IMU, Navigation, Guidance, Control, GNSS-denied, Artificial Intelligence, Deep Learning, Neural Networks, Long Short-Term Memory

I. INTRODUCTION

The recent conflicts have highlighted the vulnerability of geolocation using Global Navigation Satellite Systems (GNSS). As the GNSS signals can be jammed or spoofed, it is critical to have alternative navigation solutions. With the P-ISL (Projectile-ISL) model [1], the localization of a projectile using inertial sensors, magnetometers and Artificial Intelligence (AI) has been demonstrated: Long Short-Term Memory (LSTM) networks were trained with projectile trajectory data. Good results have been obtained mainly due to the predictable trajectory of the projectile. In this work, we propose to extend the application of this technology to all types of flying vehicles like Unmanned Aerial Vehicles (UAVs), based on [1] and on the DeepNav model [2]. To make our AI model more generic, we need to feed it with additional inputs such as control signals sent to the actuators and potentially other types of sensors.

In the literature, there exists many methods enabling navigation systems to work without GNSS thanks to Deep Learning (DL) and Neural Networks (NN). Indeed, DL and NN can be used to regress the full or the residual states of the

UAV, used to fuse sensor data with a Kalman Filter (KF) or used to filter more parameters in nonlinear filters according to Cohen et al. [3] by using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Currently, most of these navigation systems, predicting pseudo-GNSS, are hybrid systems and work with two modes, considering the availability of the GNSS signal.

II. Study Objectives

The main aim of the study is to build an AI-navigation system for Fixed-Wing UAV and low-cost sensors. To avoid spoofing attack, GNSS data are not employed. Therefore, the AI model replaces entirely the GNSS and the EKF by using additional data from other sensors and the flight commands. With those data, the AI model will predict position, velocity and attitude in quaternions format of a Fixed-Wing UAV as shown in Fig. 1.

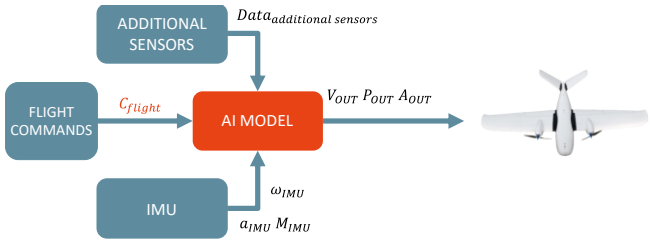


Fig. 1. Block diagram of the first objective.

III. Proposed Methodology and Work Performed

We use modeling and simulation to generate useful data of a Fixed-Wing UAV with a Matlab model to build our dataset. The Matlab model is based on the Pixhawk's Fixed-Wing attitude controller [4]. It contains a Fixed-Wing's 6 Degrees of Freedom (6DOF) model and error models of sensors as shown in Fig. 2.

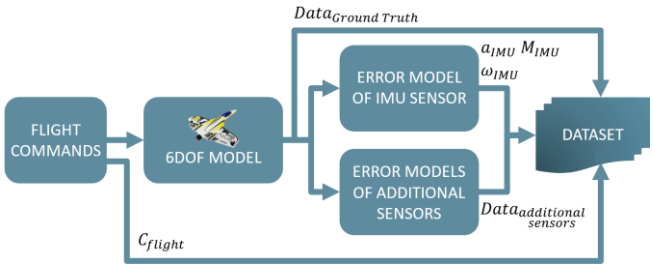


Fig. 2. Design of Matlab model.

We generated 5,000 flights containing many timeseries to build our dataset for the training of AI models. Fig. 3 shows the features used to train AI models and the labels for the prediction.

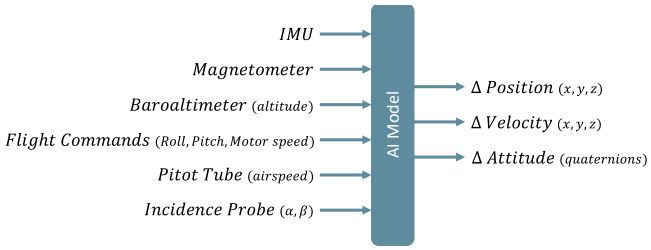


Fig. 3. Timeseries data used as features (left) and as labels (right).

In the case of our study, we compare two AI models which estimate position, velocity and attitude of an object from low-cost sensors. These models use mainly LSTM unit [5] allowing to predict timeseries at short and long term at the same time.

A test dataset containing 100 simulated flights is used with P-ISL and DeepNav. In addition, Dead-Reckoning (DR) estimation is used to be compared to the AI models. Each AI model predicts position and velocity on the XYZ axes and the attitude in quaternions format. To evaluate the performance between P-ISL and DeepNav, we employ few metrics as the Mean Average Error (MAE), the Root Mean Square Error (RMSE) and the Mean Absolute Scaled Error (MASE). According to the results, we can state that AI models have better prediction than the DR estimation. In addition, DeepNav model has better performance than P-ISL model for position estimation. However, AI models have difficulties to estimate the velocity and the attitude in quaternions format.

IV. Outlooks

To resolve those issues, we will build our AI model by using and testing other kinds of layers (recurrent layers, convolution layers and attention layers) and evaluate it with our dataset containing simulated flights. A new dataset containing simulated data with a new Fixed-Wing UAV will be generated and then to carry out real-time experiments with embedded system in a real-world setting.

References

[1] Alicia Roux, Sébastien Changey, Jonathan Weber, and Jean-Philippe Lauffenburger, "LSTM-Based Projectile Trajectory Estimation in a GNSS-Denied Environment," *MDPI Sensors*, vol. 23, no. 6, pp. 1–18, 2023, doi: https://doi.org/10.3390/s23063025.

[2] Ahmed AbdulMajuid, Osama Mohamady, Mohannad Draz, and Gamal El-bayoumi, "GPS-Denied Navigation Using Low-Cost Inertial Sensors and Recurrent Neural Networks," *Electrical Engineering and Systems Science*, pp. 1–17, 2021, doi: https://doi.org/10.48550/arXiv.2109.04861.

[3] Nadav Cohen and Itzik Klein, "Inertial Navigation Meets Deep Learning: A Survey of Current Trends and Future Directions," *arXiv Robotics*, pp. 1–15, 2023.

[4] PX4 Team, "Fixed-Wing Attitude Controller," PX4 Guide. [Online]. Available: https://docs.px4.io/main/en/flight_stack/controller_diagrams.html#fixed-wing-attitude-controller

[5] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: https://doi.org/10.1162/neco.1997.9.8.1735.

Testing AI for defence: operational needs, systemic constraints, and a sovereign evaluation platform

Agnes DELABORDE
Evaluation of AI and robotics
LNE
Paris, France
agnes.delaborde@lne.fr

Anne KALOUGUINE
Evaluation of AI and robotics
LNE
Paris, France

Rémi REGNIER
Evaluation of AI and robotics
LNE
Paris, France

Guillaume BERNARD
Evaluation of AI and robotics
LNE
Paris, France

***Abstract*—Artificial intelligence (AI) in defence must be evaluated under mission conditions where robustness, resilience and trust are critical. Beyond conventional benchmarks, effective assessment relies on pipelines combining simulation, physical trials and dataset-based testing. This paper outlines key technical and systemic challenges, and introduces the LE.IA platform developed by LNE, a sovereign infrastructure of four testbeds (Evaluation, Simulation, Immersion and Action) that delivers trustworthy, mission-relevant evaluation capabilities for defence stakeholders.**

***Index Terms*—artifial intelligence, defence, evaluation, conformity, test beds.**

Artificial intelligence in defence must be assessed under mission conditions where robustness, resilience and trust are critical. Traditional benchmarks often fail to capture the complexity of adversarial or constrained operational environments. Evaluating such systems requires mission-oriented pipelines combining simulation, physical testing, and dataset-based validation. This extended abstract discusses the main technical and systemic challenges raised by these requirements and introduces the LE.IA platform developed by LNE as a sovereign evaluation infrastructure designed to meet them.

I. Context and Challenges

The defence sector increasingly integrates AI into operational and decision-support systems, from autonomous robotics to intelligence analysis. However, these applications must perform in adversarial, uncertain, and safety-critical contexts where a single failure can have severe consequences. The European Defence Agency [1] and the US Department of Defense [2] have both highlighted the need for adaptive, lifecycle-based evaluation frameworks. Evaluating AI in this domain is not only a question of performance but of reliability, safety, explainability, and resilience under mission constraints.

Technical challenges include the limited availability of realistic data, the opacity of deep learning architectures, and the difficulty of reproducing rare yet high-consequence events. Simulation and synthetic data generation are essential but cannot fully replace testing of embedded systems under real conditions. Moreover, the lack of institutionalized Verification, Validation, Testing and Evaluation (VVT&E) cycles across the AI lifecycle remains a major structural barrier. Both RAND [3] and the EDA [1] have stressed that evaluation must be iterative and integrated throughout development, deployment and decommissioning phases, not limited to a single pre-deployment phase.

Explainability introduces another layer of complexity. Methods intended to make AI decisions transparent can themselves be manipulated through adversarial perturbations, producing deceptive or inconsistent interpretations. In defence use cases where human oversight is critical, the robustness of these interpretability techniques becomes a matter of operational safety rather than academic interest.

Systemic challenges are equally decisive. The classified nature of information restricts access to representative datasets and limits cross-border collaboration. Regulatory asymmetries between civilian and military domains complicate the design of evaluation frameworks aligned with both ethical and operational imperatives. At the same time, sovereignty concerns drive nations to maintain internal control over their evaluation capabilities, while interoperability within alliances such as NATO requires harmonized methodologies. Finally, the legitimacy of military AI relies on ethical alignment with democratic principles of accountability and proportionality. Technical reliability alone cannot ensure trust if governance mechanisms remain unclear or fragmented.

II. Use Cases and Evaluation Implications

The diversity of AI applications in defence translates into varied evaluation requirements. Three representative use cases can illustrate this need for tailored methods:

- *Computer vision for tactical awareness*: an AI model analyses live drone imagery to detect and classify threats in urban or unstructured terrain. Evaluation must integrate latency constraints, adversarial camouflage, and sensor degradation.
- *Language-based intelligence analysis*: a multilingual language model summarises intercepted communications. Testing must consider factual accuracy, cross-lingual consistency, and resistance to adversarial prompts.
- *Autonomous navigation for ground convoys*: an embedded system plans routes and reacts to obstacles under GPS denial. Evaluation must assess real-time reactivity, coordination with human operators, and recovery from failures.

Each use case requires specific test conditions and metrics. For example, vision models must be tested on diverse annotated imagery under varied weather and lighting. Language models require curated datasets reflecting semantic nuance and cultural context. Autonomous navigation must combine virtual simulation, hybrid testing and physical trials to explore edge cases and safety-critical situations. In all cases, evaluation must remain traceable, repeatable and risk-aware.

These operational examples underline several infrastructure requirements: modularity to accommodate different AI components and mission profiles; multi-layered instrumentation capturing both system-level and component-level behaviour; secure and isolated environments for classified data; and support for synthetic or simulated datasets to overcome data scarcity while preserving confidentiality.

III. The LE.IA Platform

To address these requirements, LNE has developed the LE.IA platform (*Laboratoire d'Evaluation de l'Intelligence Artificielle*, Artificial Intelligence Evaluation Laboratory), a sovereign multi-level evaluation infrastructure combining four complementary testbeds:

- *LE.IA Evaluation*, for dataset-based testing of AI software using independent and representative datasets;
- *LE.IA Simulation*, for testing within high-fidelity digital twins and synthetic environments;
- *LE.IA Immersion*, a hybrid platform where physical devices are immersed in simulated surroundings;
- *LE.IA Action*, for full physical trials of embedded AI systems in representative scenarios.

LE.IA Simulation relies on advanced rendering engines such as Unreal Engine 5 to generate realistic data and explore extreme conditions or rare events. LE.IA Immersion bridges the gap between simulation and reality by connecting physical sensors to simulated environments through real-time feedback loops. LE.IA Action extends evaluation to controlled laboratory testing of embedded AI systems under representative but delimited physical conditions, focusing on specific performance and safety aspects that complement broader operational validation.

These infrastructures are applied in ongoing projects such as COMMANDS (EDF 2023–2025) on autonomous ground vehicles, ULTIMATE (Horizon Europe 2022–2024) for hybrid AI in manufacturing and space, and KOIOS (2023–2025) on multimodal human–AI teaming. Lessons from earlier initiatives such as Blaxtair SAFE, MAURDOR and ALLIES have informed the methodological foundations of LE.IA, while initiatives such as LLMs4EU (2025-2028) extend these approaches to large language model evaluation at the European level.

IV. Mission-based engagement

As a public laboratory under the French Ministry of Economy, LNE supports national and European strategies for safety, conformity and technological trust. Its role in AI evaluation extends beyond testing: LNE co-designs evaluation frameworks with developers, authorities and standardization bodies to ensure technical soundness and regulatory alignment. The LE.IA platform is continuously adapted to emerging defence and security needs through collaborative development with institutional and industrial partners.

V. Conclusion

Evaluating AI in defence requires an integrated approach that combines operational realism, ethical accountability and sovereign control. Robust pipelines must include dataset-based, simulated, hybrid and physical testing, each complementing the others to capture the full risk spectrum. LNE's LE.IA platform embodies this principle by providing modular, mission-relevant and secure evaluation capabilities. Building sovereign infrastructures of this type is not optional: it is essential to ensure trustworthy deployment, credible oversight and strategic autonomy in the defence use of artificial intelligence.

Acknowledgment

The authors acknowledge the partial support of France 2030, Bpifrance and the European Union, notably through the PRISSMA, CitCom, TEF-Health, and agrifoodTEF projects.

References

[1] European Defence Agency, "Trustworthiness for AI in Defence," European Defence Agency, Tech. Rep., 2025, white paper. [Online]. Available: https://www.eda.europa.eu/docs/default-source/brochures/taid-white-paper-final-09052025.pdf

[2] U.S. Department of Defense, "2023 Data, Analytics, and Artificial Intelligence Adoption Strategy," Department of Defense, Tech. Rep., 2023, doD Strategy Document.

[3] RAND Corporation, "The Department of Defense's Posture for Artificial Intelligence," RAND Corporation, Tech. Rep., 2024, policy Report.

Towards Robust Military Ground Robotics

Cécile Jourdas
KNDS France Robotics
Versailles, France
cecile.jourdas@knds.fr

Bruno Ricaud
KNDS France
Versailles, France
bruno.ricaud@knds.fr

***Abstract*— This paper presents insights from over a decade developing military robotic platforms, including KNDS' Nerva platforms and Multi-Mission Tactical Robots (M2TR), designed to assist soldiers through remote-controlled and autonomous missions. Drawing from extensive field experience, we develop adaptable algorithms across diverse operational terrains and use cases. Our work demonstrates that successful autonomous military robotics requires not only advanced individual platform capabilities, but also careful consideration of field robustness, human factors, communication resilience, and system-level integration challenges inherent in transitioning from single-platform to coordinated multi-domain operations.**

Keywords—robotics, mobility, perception, tactical planning.

I. Introduction

KNDS robotics platforms assist soldiers through remote-controlled or autonomous missions, improving protection while freeing them from hazardous tasks. The primary challenge lies in the variety of use cases and operational terrains, requiring adaptable, robust algorithms capable of generalizing across diverse scenarios. Each platform incorporates multiple sensors alongside sophisticated autonomous behaviors coordinated for complex multi-platform missions.

Our main contributions in this paper are, first, identifying the sensors, functions, and behaviors required for operational credibility and integration into multi-platform autonomous unit, then, providing field experimentations feedbacks highlighting the need for reliable communication, localization systems, and HMI ergonomics and finally outlining envisioned tactical missions.

II. Towards building trust and robustness

Robotics faces challenges in perception, mobility, communication, localization, and human-robot collaboration. Beyond advancing core perception and mobility algorithms, establishing trust in autonomous systems demands extensive user familiarization and repeated demonstrations of platform performance and robustness to facilitate their future operational integration.

A. Robot platforms

KNDS develops several robotics platforms. KNDS' Nerva platforms are modular lightweight Unmanned Ground Vehicles (UGV) configurable for CBRN detection, reconnaissance, surveillance, and counter-IED operations. M2TRs (KNDS' ULTRO carrier, KNDS' OPTIO tracked combat robot, KNDS' CENTURIO multipurpose) support various missions and with larger calculators facilitate state-of-the-art algorithm integration, though their size makes real-life testing more complex.

B. Data and simulation

Limited accessibility to operational field data is a critical issue. Creating digital twins enables early function verification, while Hardware-in-the-Loop (HIL) simulators, such as 4D Virtualiz [1] simulator, shorten integration time by providing realistic sensor data. Additionally, photorealistic simulators, such as Unreal Engine [2] and AIVerse [3], may provide specific true-to-life training data imaging.

C. Perception

Perception is critical for robotic platforms to understand and safely navigate their environments. Our work therefore focuses on mono-sensor algorithms and multi-sensor fusion (cameras, LIDAR, audio…) to enhance situational and tactical awareness and distinguish terrain types, obstacles, and key features. Satellite or aerial semantic analysis additionally help assess terrain navigability before deployment, while LIDAR data enables detailed 3D reconstructions and real-time traversability estimation.

D. Mobility

Autonomous platforms must adapt to their environment and robust multi-sensor perception is essential for dependable mobility. In off-road conditions, for instance, even tall grass may trigger false obstacles, requiring adaptive collision-avoidance settings. Ultimately, achieving true autonomy, where robots can be tasked like armored crews or infantry units, demands a dependable "Go To" function capable of operating in all terrains and weather conditions.

III. Towards experiments and missions

A. Communication and localization

The Cohoma challenge [4], organized by the French Army's BattleLab Terre, aims to integrate unmanned ground and aerial systems for tactical missions on realistic terrains with minimal operators. Key findings emphasize that successful systems require more than core platforms.

Meshed communication links between HMIs and platforms are necessary, though bandwidth must be shared among all video feeds. Communication issues stem from limitations on steep, vegetation-covered terrain and highlight the need for degraded-mode functions when links are lost, preventing uncontrolled movement or capture.

Equally vital is dependable localization. Because GPS is easily jammed, both active and passive backup methods are needed. Active positioning often relies on LIDAR or Radar SLAM for accuracy, while visual odometry and visual relocalization provide passive alternatives to maintain spatial awareness under GPS-denied conditions.

B. Defining an autonomy level

Assessing and categorizing automation levels in robotic systems is complex but essential to distinguish remote-controlled systems from those with autonomous capabilities. Operational users have to know if a robot can patrol a military camp in various conditions (day/night, clear/rainy weather, mud/dust challenges) and whether it demonstrates maturity across different contexts. Several frameworks exist but none fully address these needs and a new scale is required, expressing operational deployment capability, incorporating environmental/mission criteria and operational perspective, while remaining easily interpretable.

C. Integration in multi-platforms systems

Operators cannot manage multiple interfaces due to ergonomic limitations. The ideal solution consolidates all HMIs into a unified system controlling any platform, though determining the standard interface and responsible developer presents significant challenges. In the Cohoma challenge, limited resources led us to implement a tactical layer using ATAK (Android Team Awareness Kit) [6] with dedicated plugins to centralize mission data and control all UAV and UGV autonomous functions. This simplified interface allowed operators to manage any platform without mastering its native controls, supporting wider field acceptance through simplicity and autonomy.

To anticipate integration challenges and operational complexity, simulation-based design is essential. KNDS France's OUISARD methodology [8], based on the Wizard of Oz principle [10], enables immersive testing of Human-Robot collaboration in realistic scenarios. It helps identify user needs early, refine organizational structures, and optimize interface design. Used for tactical training and vehicle concept studies, OUISARD also gathers human factors data to guide capability development and ensure effective integration of robotic systems.

IV. Mission planing

A. Tactical flight

The use of UGVs/UAVs in modern warfare has revealed extremely high attrition rates, sometimes exceeding 75% in Ukraine [9]. To improve aerial system survivability, we digitize French Army light aviation tactical flight techniques and exploit terrain features (depressions, vegetation, structures) to reduce detection probability.

Our approach is built on three interconnected layers. First, the terrain analysis extracts key environmental features (concealment areas, masking zones, approach corridors). Then, the mission planning designs tactical trajectories and sequencing actions that balance exposure, sensor use, and mission efficiency. Finally, real-time adaptations to unexpected changes are enabled by rapid and often autonomous decision making.

The ultimate goal is indeed to enhance UAV/UGV survivability and operational efficiency through terrain-aware adaptive autonomy while preserving human control and tactical intent.

B. Site surveillance with heterogenous swarm

Planning coordinated movement of a platforms group requires high levels of embedded autonomy, as well as decentralized decision-making capabilities to handle degraded operational modes induced by communication or localization failures.

The SERBERE RAPID project (with MASA Group [7]) proved that an ideal tactical system could effectively manage a heterogeneous platform swarm using only the constrained, localized environmental knowledge that individual platforms actually possessed. Autonomous individual behaviors included patrolling with obstacle avoidance, battery management, localization recovery, and target tracking, while coordinated missions such as area coverage and relay operations were also developed.

V. Conclusion

This article has illustrated the efforts undertaken by KNDS France to make ground military robotics a reality by adopting a system-level perspective. Without this systemic approach rooted in decades of experience in designing complex ground combat systems, military ground robotic platforms risk remaining mere technological gadgets, unable to deliver their full operational potential on the battlefield.

Indeed, military ground robotics represents a particularly demanding challenge as it requires constant innovation to overcome the difficulties of self-functioning in endangered environments. However, its successful integration depends on maintaining simplicity and robustness both at the technical level and in the design of effective and intuitive Human-System collaboration.

It is only through this dual requirement, i.e. technological sophistication and operational pragmatism, that military ground robotic systems will become real and fully integrated as a mission-relevant asset for modern armed forces.

References

[1] 4D-Virtualiz, "Simulation immersive 4D pour la défense et l'industrie", *Fiche entreprise*, 2023. https://4d-virtualiz.fr/

[2] Epic Games, "Unreal Engine: Simulation, Training, and Defense", *White Paper*, 2022. https://www.unrealengine.com/en-US/solutions/training-simulation

[3] AIVerse, "La simulation IA en environnement virtuel pour l'entraînement tactique", *Présentation entreprise*, 2023. https://aiverse.io/

[4] Ministère des Armées, "Challenge COHOMA – Combat Haute Intensité et MAniabilité", *BattleLab Terre*, 2022.

[5] H. L. Choi, M. J. Kim, "Electronic Warfare in Ukraine: Lessons for Communications and ISR", *RAND Corporation*, 2023.

[6] ATAK, "Android Team Awareness Kit (ATAK) – Tactical Situational Awareness", *US Department of Defense*, 2021. https://atak.gov/

[7] MASA Group, "Simulation, IA et entraînement : Solutions MASA pour la défense", *Brochure MASA*, 2022. https://www.masagroup.net/

[8] B. Ricaud, L. Kujawa, M. Durandeau. « Imaginer et concevoir les interactions Humain-Robot du futur : une application à la robotisation du champ de bataille. », IHM'23 - 34e Conférence Internationale Francophone sur l'Interaction Humain-Machine, 2023.

[9] C. Cancian and S. Frederick, "Drone Saturation: Russia's Shahed Campaign," Center for Strategic and International Studies (CSIS), 2024.

[10] Steinfeld A., Jenkins C. J., & Scassellati B. (2009). *The Oz of Wizard: Simulating the Human for Interaction Research*. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI 2009)*, La Jolla, CA.

A Two-Stage Radar, Optronic and AIS Track-to-Track Fusion Algorithm for Enhanced Maritime Surveillance and Navigation

Cédric Seren
Maritime Environment Perception Dept.
SIREHNA/NAVAL Group
Nantes, France
cedric.seren@sirehna.com
https://orcid.org/0000-0001-5798-6710

Marie-Jeanne Paul
Maritime Environment Perception Dept.
SIREHNA/NAVAL Group
Nantes, France
marie-jeanne.paul@sirehna.com

Sylvain Vandernotte
Maritime Environment Perception Dept.
SIREHNA/NAVAL Group
Nantes, France
sylvain.vandernotte@sirehna.com

***Abstract*—In the context of autonomous navigation of Unmanned Surface Vehicles (USVs), the recent development of maritime optronic surveillance systems, boosted by the abundance of deep learning models for sea surface or aerial targets detection, raises a need for information fusion algorithms to be developed. Used as an additional dissimilar sensor channel that is helpful for establishing vessel's navigational situation awareness, the information provided by this new and emerging technology must be consolidated with that available from radar based tracking algorithms and Automatic Identification System (AIS) as a prerequisite for the development of high-performing autonomous navigation functions. To this aim, this paper introduces a two-stage radar/optronic Track-to-Track (T2T) fusion and AIS Data Association (DA) algorithm based on multiple hypothesis testing and a direct application of the Dezert-Smarandache Theory (DSmT) to assess the quality of AIS measurements for updating the fused or local target tracks. The proposed algorithm has been both tested at sea and evaluated on real sea trial data.**

***Index Terms*—Multi-Target Tracking, Track-to-Track Fusion, Data Association, Radar, Optronic, AIS, Situational Awareness**

I. Industrial Context

In maritime surveillance and navigation, ship's situational awareness at sea relies commonly on several decentralized Multi-Target Tracking (MTT) algorithms attached to one specific type of sensor (radar, camera, AIS, etc.). These latter provide a set of potentially heterogeneous tracks related to all detected sea-surface or aerial objects in observed 3D scene. Such elaborated information plays a pivotal role to:

- assist and guide every decision made by ship crew w.r.t. the encountered maritime situation (e.g., marine traffic mapping and monitoring for ship navigation, asymmetric threats detection and localization for military operation);
- perform safe maritime navigation in line with the International Regulations for Preventing Collisions at Sea [1] (known as COLREGS, for COLlision REGulationS);
- or enable fully autonomous operations of USVs as MTT serves as a fundamental aspect of situational awareness for intelligent systems in marine industry (see Fig. 1).

Nevertheless, the possible tracks diversity argues for the development of multitarget/multisensor T2T fusion algorithms, despite their suboptimal nature, to improve the overall readibility of maritime situational awareness and, at the same time, reduce the global uncertainty attached with objects recognition and localization. To these aims, many T2T solutions can therefore be designed to estimate a consolidated surrounding maritime situation useful for USVs autonomy, teleoperator or even ship crew on board. For many years now, SIREHNA has extensively investigated this research field as a naval systems equipment manufacturer for both civil and military markets. SIREHNA has developed several products such as BRIZO®,[1] and MOS®,[2] that realize sensor-level objects tracking tasks. Backed by this expertise, SIREHNA is now looking to develop its own T2T fusion embedded system in order to merge in together previous radar and optronic tracks information **[Stage ❶]** supplemented by AIS measurements when available that will be taken into account by solving a standard Data Association (DA) problem **[Stage ❷]**. This article presents an original two-stage algorithm for radar/optronic T2T fusion and AIS DA which combines: (i) a multiple hypothesis testing to check the statistical consistency between all local track estimates provided by both BRIZO® and MOS® subsystems; with: (ii) an application of the DSmT through the use of the Quality Assessment Data Association (QADA) algorithm for AIS measurements association.

Fig. 1. Seaquest-S USV (SIREHNA/NAVAL Group).

[1] BRIZO is a radar-based collision avoidance system offering automatic detection, tracking and geolocation of any obstacles.

[2] MOS: Maritime Optronic Surveillance system based on AI object detection in images.

II. Proposed Algorithm: Two-Stage Track-to-Track Fusion and Data Association

The method developed in this paper separates the multi-target/multisensor data fusion into two coupled procedures due to the specific processing of AIS data. Fig. 2 depicts the flowchart of the proposed methodology. It is noteworthy that such layout permits to obtain fused tracks of different nature: radar/optronic/AIS, radar/AIS, radar/optronic, optronic/AIS namely, but also to retrieve the local track state estimates if no concordance and association have been declared.

In stage one, the agreement between two tracks is declared as soon as a multiple hypothesis testing is satisfied, taking into account an estimation of the cross-correlation between the considered tracks, leading to apply the following *cross-covariance* equations [2] as the T2T fusion formulas:

$$\left\{ \begin{array}{c} \hat{\mathbf{x}} = \hat{\mathbf{x}}_i + \mathbf{K}\,(\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_i) \;\text{ where: }\; \mathbf{K} = (\widehat{\mathbf{P}}_i - \widehat{\mathbf{P}}_{ij})\boldsymbol{\mathcal{P}}_{ij}^{-1} \\ \text{and: } \boldsymbol{\mathcal{P}}_{ij} = \widehat{\mathbf{P}}_i + \widehat{\mathbf{P}}_j - \widehat{\mathbf{P}}_{ij} - \widehat{\mathbf{P}}_{ij}^T \end{array} \right. \quad (1)$$

Before achieving such T2T fusion between a radar track and an optronic track, it appears mandatory to assess if two local tracks, issued from two distinct sensors and characterized by their respective statistics $(\hat{\mathbf{x}}_i, \widehat{\mathbf{P}}_i)$ and $(\hat{\mathbf{x}}_j, \widehat{\mathbf{P}}_j)$, correspond or not to the same target. To this aim, we define:

$$\widehat{\boldsymbol{\Delta}}_{ij}(k|\cdot) = \hat{\mathbf{x}}_i(k|\cdot) - \hat{\mathbf{x}}_j(k|\cdot) \in \mathbb{R}^{n_{\mathbf{x}}} \quad (2)$$

the estimate of the difference $\boldsymbol{\Delta}_{ij}(k) = \mathbf{x}_i(k) - \mathbf{x}_j(k)$ where $\mathbf{x}_i$ and $\mathbf{x}_j$ designate the true states of the targets. The *instantaneous concordance test*[3] (aka association or correlation test) between the two local tracks i and j is based on the Mahalanobis distance d defined by (assuming normally distributed local estimation errors):

$$d \triangleq \widehat{\boldsymbol{\Delta}}_{ij}^T(k|\cdot)\mathbf{T}_{ij}^{-1}(k|\cdot)\,\widehat{\boldsymbol{\Delta}}_{ij}(k|\cdot) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \delta \quad (3)$$

$$\begin{aligned} \text{where: } \mathbf{T}_{ij}(k|\cdot) &= E[\widetilde{\boldsymbol{\Delta}}_{ij}(k|\cdot)\,\widetilde{\boldsymbol{\Delta}}_{ij}^T(k|\cdot)] \\ &= \widehat{\mathbf{P}}_i(k|\cdot) + \widehat{\mathbf{P}}_j(k|\cdot) - \widehat{\mathbf{P}}_{ij}(k|\cdot) - \widehat{\mathbf{P}}_{ij}^T(k|\cdot) \\ \text{and: } \widetilde{\boldsymbol{\Delta}}_{ij}(k|\cdot) &\triangleq \boldsymbol{\Delta}_{ij}(k) - \widehat{\boldsymbol{\Delta}}_{ij}(k|\cdot) \end{aligned}$$

The notation $E[\,\cdot\,]$ refers to the expectation operator. In (3), threshold δ is such that:

$$P\{d > \delta | \,\mathcal{H}_0\} = \alpha \quad (4)$$

where α is a given value defined a priori. Selecting a threshold value for variable δ can rely on the assumption that $\boldsymbol{\Delta}_{ij}(k)$ must follow a normal law in case of concordance. Under this reasoning, d must therefore be distributed as a χ^2 law (as a weighted sum of squares of $n_{\mathbf{x}}$ normal laws) with $n_{\mathbf{x}}$ degrees of freedom. These equations involve an estimation of the cross-covariance matrix $\widehat{\mathbf{P}}_{ij}$ between tracks $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ which can be approximated by the Hadamard product, denoted by $\odot$ in the sequel, of both local covariance matrices [3] i.e.:

$$\widehat{\mathbf{P}}_{ij} = \rho\sqrt{\widehat{\mathbf{P}}_i \odot \widehat{\mathbf{P}}_j} \text{ with: } \rho \simeq 0.4 \quad (5)$$

[3] Hypothesis $\mathcal{H}_0$ stands for tracks agreement while $\mathcal{H}_1$ is for tracks mismatch.

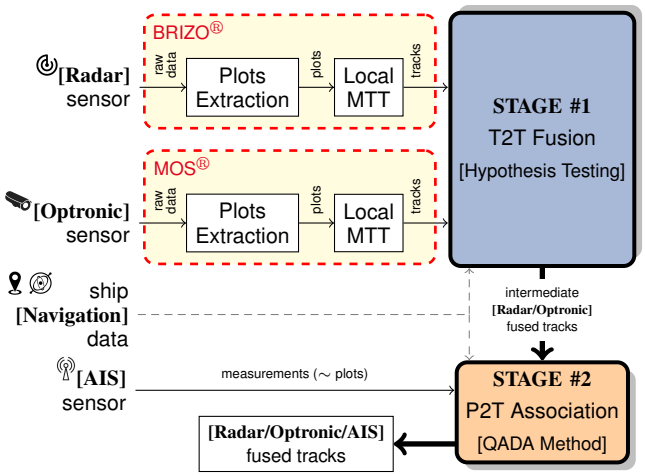


Fig. 2. Flowchart of the proposed two-stage algorithm.

The main drawback of such an approach lies in its relatively high sensitivity to local tracking filters consistency that must be guaranteed for working properly. To circumvent this weakness in the method and robustify the whole procedure, an extra test based on the Bhattacharyya distance which measures the similarity between two probability distributions [4] was added.

The purpose of stage two is to increase the information contained in all available tracks, whatever these latter correspond to fused or local state estimate, by seeking any possible Plot-to-Track (P2T) association with available AIS measurements $\mathbf{z}_l,\ l \in [\![\,1\,;n_m\,]\!]$. To this aim, the QADA method based on the DSmT [5] is applied and allows to update some track estimates in line with the confidence level $q(l,m) \in [\,0\,;1\,]$ calculated by the method (and aka quality indicator) that characterize each P2T association. When all the indicator values have been determined, the states $\hat{\mathbf{x}}_m,\ m \in [\![\,1\,;n_t\,]\!]$ of the associated tracks are updated by the AIS information according to the filtering equation:

$$\hat{\mathbf{x}}_m(k|k) = \hat{\mathbf{x}}_m(k|k-1) + \mathbf{K}(k)\,(\mathbf{z}_l(k) - \hat{\mathbf{z}}_m(k|k-1)) \quad (6)$$

where the gain correction matrix now reads for every pairing:

$$\mathbf{K}(k) = \widehat{\mathbf{P}}(k|k-1)\mathbf{C}^T(k)\left[\mathbf{C}(k)\widehat{\mathbf{P}}(k|k-1)\mathbf{C}^T(k) + \frac{\mathbf{R}_{\text{AIS}}(k)}{q(l,m)}\right]^{-1}$$

Asymptotically, we have for each association $(\mathbf{z}_l, \hat{\mathbf{x}}_m)$:

- trustful ▶ $q(l,m) \mapsto 1^- \Rightarrow\ \mathbf{R}_{\text{AIS}}(k)$ left unchanged;
- distrustful ▶ $q(l,m) \mapsto 0^+ \Rightarrow\ \mathbf{R}_{\text{AIS}}(k)$ increased.

References

[1] IMO, "COLREG: Convention on the International Regulations for Preventing Collisions at Sea," International Maritime Organization, 2003.

[2] Y. Bar-Shalom, "On the track-to-track correlation problem," IEEE Trans. Autom. Control, vol. AC-26, no. 2, pp. 571–572, 1981.

[3] S. Matzka and R. Altendorfer, "A comparison of track-to-track fusion algorithms for automotive sensor fusion," In Proc. IEEE Intl. Conf. Multisensor Fusion Integr. Intell. Syst., pp. 189–194, 2008.

[4] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," Sankhyā. vol. 7(4), JSTOR 25047882, 1946.

[5] J. Dezert, A. Tchamova, P. Konstantinova and E. Blasch, "A comparative analysis of QADA-KF with JPDAF for multitarget tracking in clutter," In Proc. of the 20th Intl. Conf. on Infor. Fusion, 2017.