42nd International Symposium on Military Operational Research (42 ISMOR)

Robust Machine Learning for Naval Image Classification in the Blue Amazon

Gabriel Custódio Rangel [a], Luiz Frederico H. S. B.Teixeira [a], Victor Benicio A. S. Alves, Igor Pinheiro de A. Costa [a,], Johannes O. Royset [b], Eric C. Eckstrand [c]

[a] *Naval Systems Analysis Center (CASNAV), Rio de Janeiro, RJ, Brazil*

[b] *University of Southern California (USC), Los Angeles, CA, EUA*

[c] *Naval Postgraduate School (NPS), Monterey, CA, EUA*

BLUF

- Propose a robust machine learning algorithm using maritime computer vision for surveillance in areas at risk of adversarial attacks in the Blue Amazon. The algorithm will address a binary classification problem, detecting the presence or absence of ships in these regions to feed the monitoring system of large coastal areas, even under a certain degree of label noise inherent to the data.

Introduction

Ship classification and detection systems are employed in various contexts where they may pose challenges, including:

- National and local defense;
- Safeguard busy coastal and strait passages;
- Ship traffic management;
- Combating illegal fishing;
- Piracy;
- Drug smuggling;
- Human trafficking; and
- Monitoring the global trade network

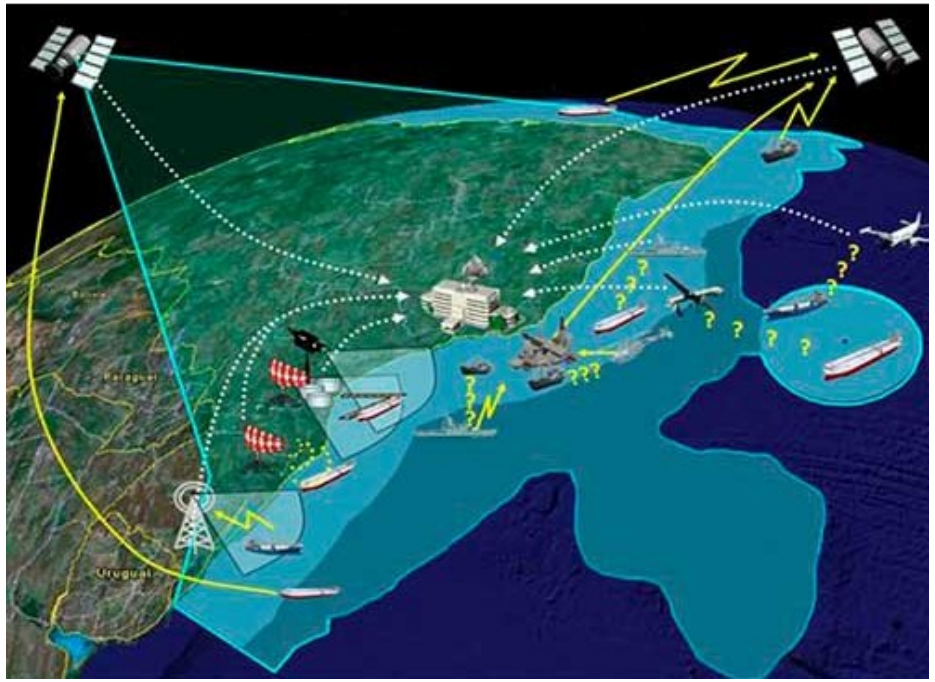
Ship traffic density

Introduction

This study explores the advantages of utilizing alternative Machine Learning algorithms with imagery datasets to improve the ship detection system, which is utilized for safeguarding territorial waters. These models will provide supplementary capabilities to discourage unlawful activities in those regions.

Background

- National and local defense;
- Green x Blue Amazon
- Satellite Imagery
- Tracking System



Blue Amazon Management System (SisGAAz)

Problem Formulation

The optimization problem we are solving has the following form

$$\underset{w\in\mathbb{R}^d, u\in U}{\text{minimize}} \ \sum_{j=1}^{n} \left((p_j + u_j) f_j(w) + \theta \left| u_j \right|\right)$$

- θ represents a penalty parameter,
- w is a d-dimensional vector of neural net weights, and
- u is a perturbation n-dimensional vector that alters the nominal probabilities $p = (p_1, \ldots, p_n)$, usually with each value p_j equal to $1/n$.
- For a given w, $f_j(w)$ gives the loss associated with data point j.

$$U = \left\{ u \in \mathbb{R}^n \middle| u_j \geq -p_j, \qquad \sum_{j=1}^{n} u_j = 0 \right\}$$

Methodology

We adopt Rockafellian risk minimization introduced by Royset et al. 2022.

This method consists of the usage of the Alternating Direction Heuristic (ADH), which alternates between optimizing different sets of variables while keeping the others fixed, hence the name "alternating direction". First, we optimize the w

$$\underset{w\in\mathbb{R}^d}{\text{minimize}} \ \sum_{j=1}^{n} \ p_j^v f_j(w)$$

Then we use a subroutine to optimize over u for a fixed w

$$\underset{v\in\mathbb{R}^n, u\in U}{\text{minimize}} \ \sum_{j=1}^{n} \ (u_j f_j(w^v) + \theta v_j)$$

Keep alternating between the optimizations until both iterations are completed

Rockafellian Relaxation in Optimization under Uncertainty: Asymptotically Exact Formulations
LL Chen, JO Royset
arXiv preprint arXiv:2204.04762

U update methods

- The *u*-values act as weights that indicate the importance of observation. A negative *u*-value reduces the importance during training, while a positive u-value increases it.

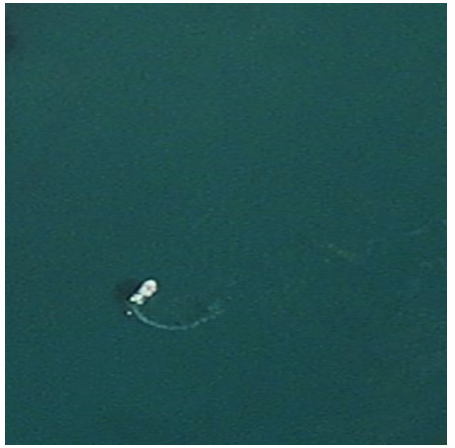
- To solve the optimization problem using this methodology and perform the *u* update, we used the following alternatives:
  - No update ;
    - All *u* values are set to zero, and the model's accuracy is equivalent to Empirical Risk Minimization (ERM).
  - Rockafellian Risk Minimization (RRM)
    - Linear Programming-Based Update;
      Solve the linear optimization using Pyomo, to generate the *u* values
    - Subgradient Update.
      Define a step size $\lambda > 0$, number of sub-iterations, and compute the penalties associated to data points readjusting the *u* values.

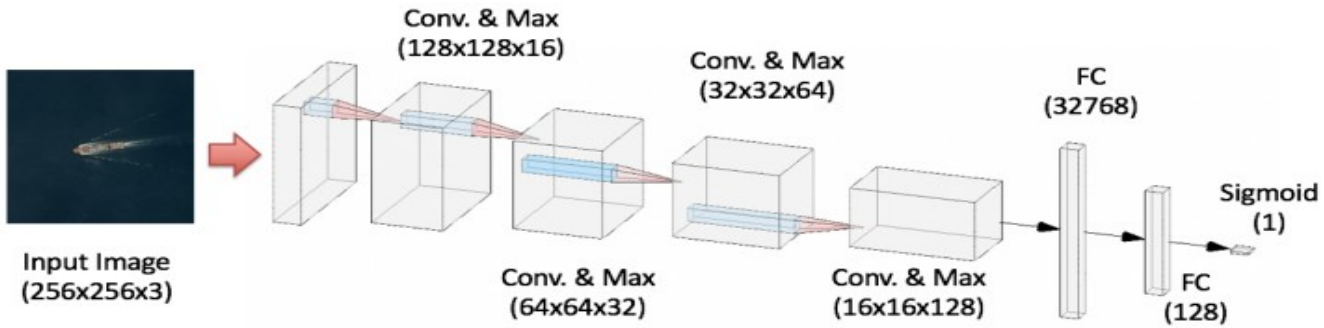
Overview

- Machine Learning Concepts;
- Binary Classification Problem;
- Convolutional Neural Networks
- 2 optimizers – SGD and ADAM

No ship (Class 0)

Ship (Class 1)





python™ NumPy matplotlib

TensorFlow scikit learn K Keras

Processed in Hamming Cluster

The Hamming (GPU) nodes, provide visualization and graphical processing of research data sets.

Open-Source Datasets

Datasets

MASATI v2 (Maritime Satellite Imagery)

- This dataset provides maritime scenes of optical aerial images from the visible spectrum and can be used to evaluate ship detection methods. Each image may contain one or multiple targets in different weather and illumination conditions.
- Image size has a spatial resolution of 512 x 512 pixels. The images are stored as PNG, where pixel values represent RGB colors.

- Images are resized to 128x128x3

in the data pre-processing

Sample of Dataset:

- 1022 images of “no ship” and
- 1027 images of ships

MASATI v2 (Maritime Satellite Imagery)

Datasets

Airbus Ship Detection

- We obtained a public dataset on the Challenge website (Kaggle, 2018).
- The dataset image resolution is 768 × 768 pixels. The images are stored as JPG, where pixel values represent RGB colors.
- Images are resized to 128x128x3 in the data pre-processing

Sample of Dataset:

- 5214 images of “no ship” and
- 5214 images of ships

Airbus Ship Detection

Description of the CNN used for Adam

#	Layer	Filters	Kernel	Output Size	# Parameters
1	Convolution	16	3x3	128 x 128 x 16	448
	Max-Pooling		2x2	64 x 64 x 16	
2	Convolution	32	3x3	64 x 64 x 32	4,640
	Max-Pooling		2x2	32 x 32 x 32	
3	Convolution	64	3x3	32 x 32 x 64	18,496
	Max-Pooling		2x2	16 x 16 x 64	
4	Convolution	128	3x3	16 x 16 x 128	73,856
	Max-Pooling		2x2	8 x 8 x 128	
5	Fully-connected	256		1 x 256	2,097,408
6	Fully-connected	128		1 x 128	32,896
7	Softmax (Output)	2		1 x 2	258

Description of the CNN used for SGD

#	Layer	Filters	Kernel Size	Output Size	# Parameters
1	Convolution	32	3x3	128 x 128 x 32	896
2	Batch Normalization Activation				128
3	Max-Pooling		2x2	64 x 64 x 32	-
4	Convolution Activation	64	3x3	64 x 64 x 64	18,496
	Max-Pooling		2x2	32 x 32 x 64	
5	Fully-connected	128		1 x 128	8,388,736
6	Batch Normalization Activation				512
7	Softmax (Output)	2		1 x 2	258

Next Step

Label noise occurs when the labels or outcomes of data points are incorrect because they may have been distorted or altered from their original, accurate labels. Notably, in real-world datasets, the prevalence of such corrupted labels is reported to vary, with estimates ranging between 8.0% and 38.5%.

Data corruption can occur due to various reasons such as data collection and labelling entry errors, or even adversarial attacks intended to poison the training data and disrupt the system

RRM aims to enhance the resilience of machine learning models to training data that may be corrupted, such as in cases where the labels are incorrect.

Hyper Parameters for ERM and RRM

Algorithm	Parameters				
	Number of epochs (κ)	Number of iterations (τ)	Stepsize (μ or λ)	Sub-iterations (σ)	Penalty (θ)
ERM	500	1	-	-	-
RRM(ADH-LP)	10	50	0.5	-	0.15, 0.20, 0.25, 0.30, 0.35
RRM(ADH-SUB)	10	50	0.3	10	

We first evaluate the models using a 0% label contamination training set.

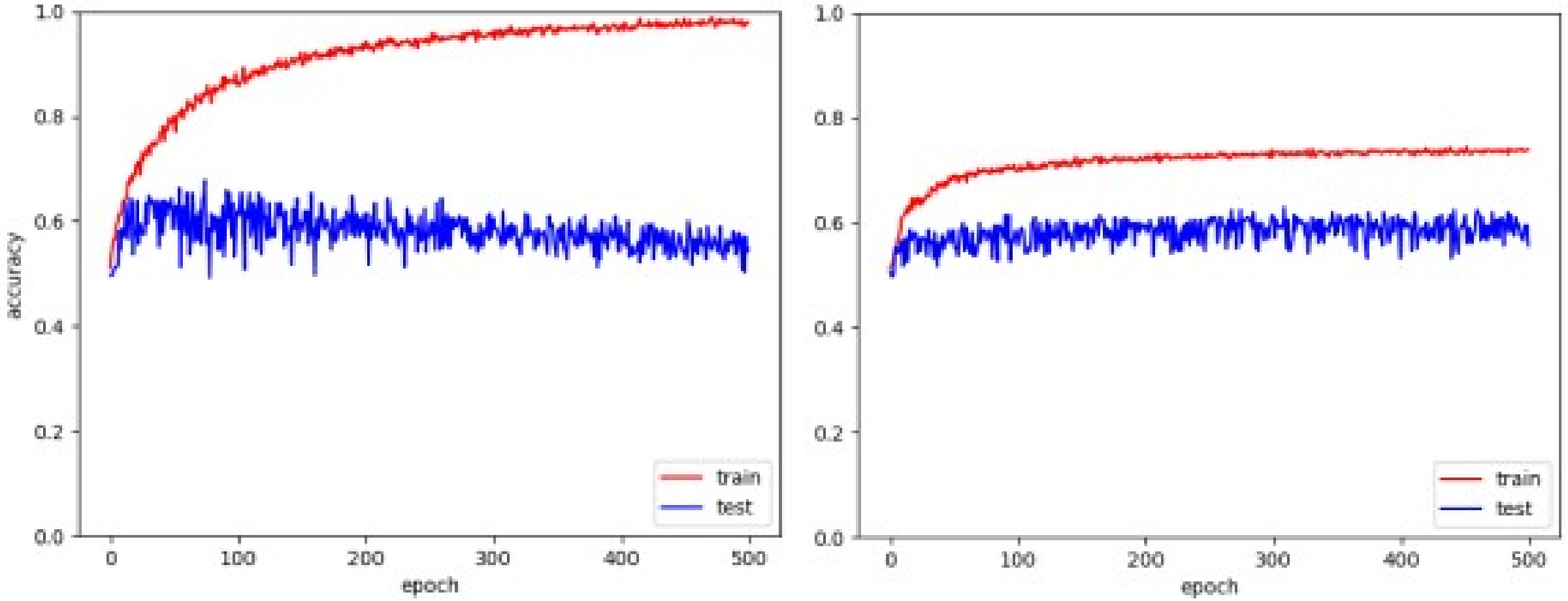
Then we gradually introduce higher levels of contamination, such as 10%, 20%, 30%, and 40%. This evaluation can be carried out for both the MASATI and Airbus models to assess their robustness to label contamination.

Final test accuracy in MASATI for ERM and RRM (ADH-LP / SGD)

Method	Corrupted training data percentage				
	40%	30%	20%	10%	0%
ERM	0.556	0.546	0.581	0.663	0.648
RRM ($\mu = 0.5$)					
$\theta = 0.15$	0.604	0.648	0.648	0.639	0.634
$\theta = 0.20$	0.663	0.692	0.648	0.648	0.609
$\theta = 0.25$	0.639	0.687	0.658	0.629	0.614
$\theta = 0.30$	0.668	0.585	0.600	0.643	0.634
$\theta = 0.35$	0.624	0.556	0.639	0.648	0.653

The values highlighted in gray represent the cases where RRM outperform or match ERM.

Training and test accuracy for ERM (left) and RRM ADH-LP/ $\theta = 0.35$ (right) on MASATI with 30% of contamination



Evolution of u-vector across ADH-LP / SGD in MASATI (30% of contamination and θ = 0.35)

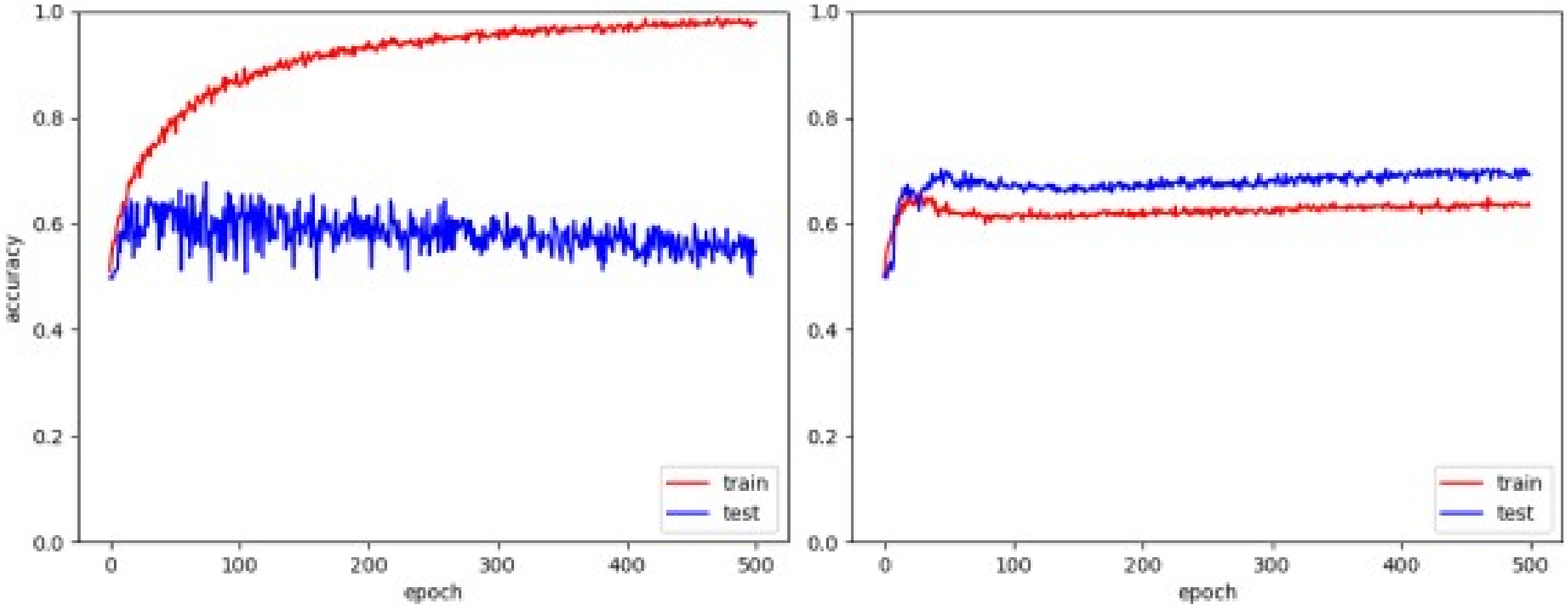
Nominal probability (1/N)	Iteration number					
	i=1		*i*=2		*i*=49	
$5.4 \cdot 10^{-4}$ u_l-values	mislabeled images	correct labeled images	mislabeled images	correct labeled images	mislabeled images	correct labeled images
>>0	0	1	0	1	1	1
≈ 0	304	813	266	782	338	1,005
$-1.5 \cdot 10^{-4}$	0	0	37	92	1	5
$-2.7 \cdot 10^{-4}$	249	477	38	41	0	1
$-4.0 \cdot 10^{-4}$	0	0	212	375	0	1
$-5.4 \cdot 10^{-4}$	0	0	0	0	213	278
Total of images	553	1,291	553	1,291	553	1,291

Final test accuracy in MASATI for ERM and RRM (ADH-LP / SGD)

Method	Corrupted training data percentage				
	40%	30%	20%	10%	0%
ERM	0.556	0.546	0.581	0.663	0.648
RRM ($\mu = 0.5$)					
$\theta = 0.15$	0.604	0.648	0.648	0.639	0.634
$\theta = 0.20$	0.663	0.692	0.648	0.648	0.609
$\theta = 0.25$	0.639	0.687	0.658	0.629	0.614
$\theta = 0.30$	0.668	0.585	0.600	0.643	0.634
$\theta = 0.35$	0.624	0.556	0.639	0.648	0.653

The values highlighted in gray represent the cases where RRM outperform or match ERM.

Training and test accuracy for ERM (left) and RRM ADH-LP/θ =0.20 (right) on MASATI with 30% of contamination



Evolution of u-vector across ADH-LP / SGD in MASATI (30% of contamination and θ = 0.20)

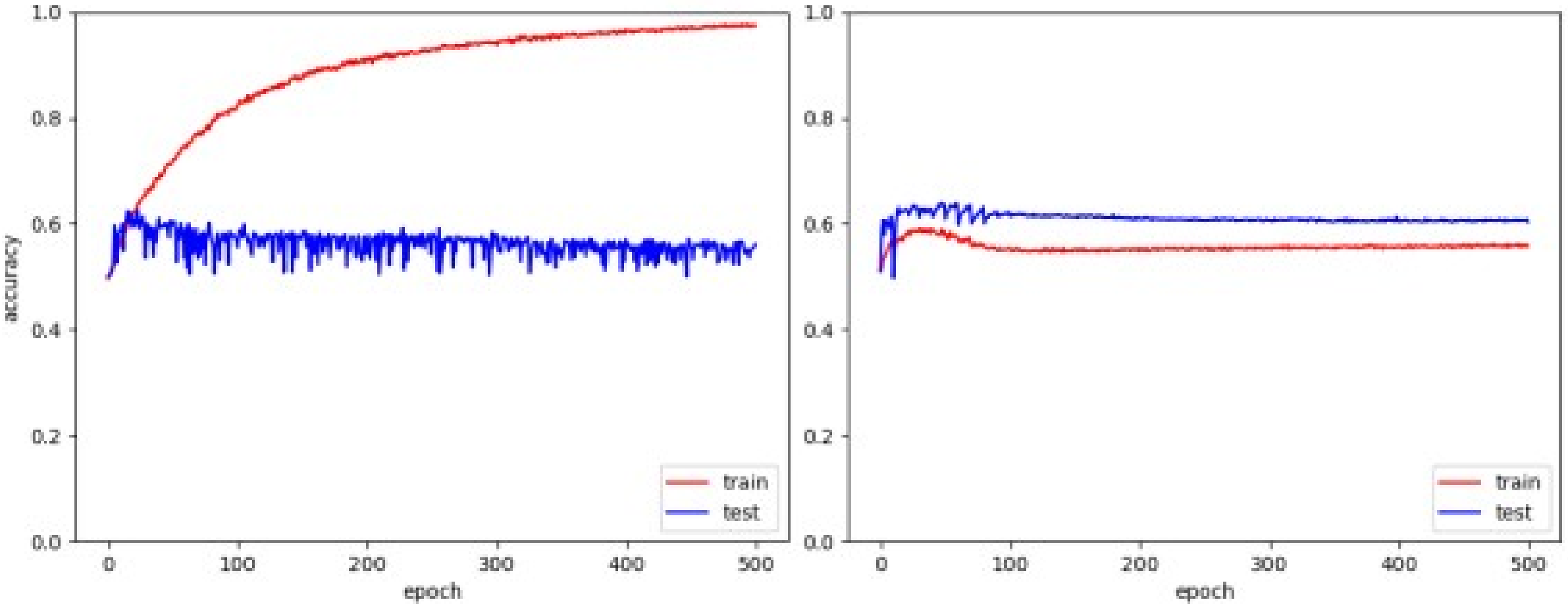
Nominal probability (1/N) $5.4 \cdot 10^{-4}$ u_i-values	Iteration number					
	$i = 1$		$i = 2$		$i = 49$	
	mislabeled images	correct labeled images	mislabeled images	correct labeled images	mislabeled images	correct labeled images
>>0	0	1	0	1	0	1
≈ 0	133	453	93	370	260	863
$-1.5 \cdot 10^{-4}$	0	0	39	105	3	1
$-2.7 \cdot 10^{-4}$	420	837	40	83	0	0
$-4.0 \cdot 10^{-4}$	0	0	381	732	0	0
$-5.4 \cdot 10^{-4}$	0	0	0	0	290	426
Total of labels	553	1,291	553	1,291	553	1,291

Final test accuracy in AIRBUS for ERM and RRM (ADH-LP / SGD)

Method	Corrupted training data percentage				
	40%	30%	20%	10%	0%
ERM	0.560	0.629	0.681	0.735	0.769
RRM ($\mu = 0.5$)					
$\theta = 0.15$	0.603	0.739	0.745	0.774	0.764
$\theta = 0.20$	0.671	0.755	0.753	0.775	0.769
$\theta = 0.25$	0.687	0.733	0.757	0.769	0.767
$\theta = 0.30$	0.684	0.744	0.764	0.765	0.774
$\theta = 0.35$	0.661	0.747	0.764	0.770	0.779

The values highlighted in gray represent the cases where RRM outperform or match ERM.

Training and test accuracy for ERM (left) and RRM ADH-LP/θ =0.15 (right) on AIRBUS with 40% of contamination



Evolution of u-vector across ADH-LP / SGD in AIRBUS (40% of contamination and θ = 0.15)

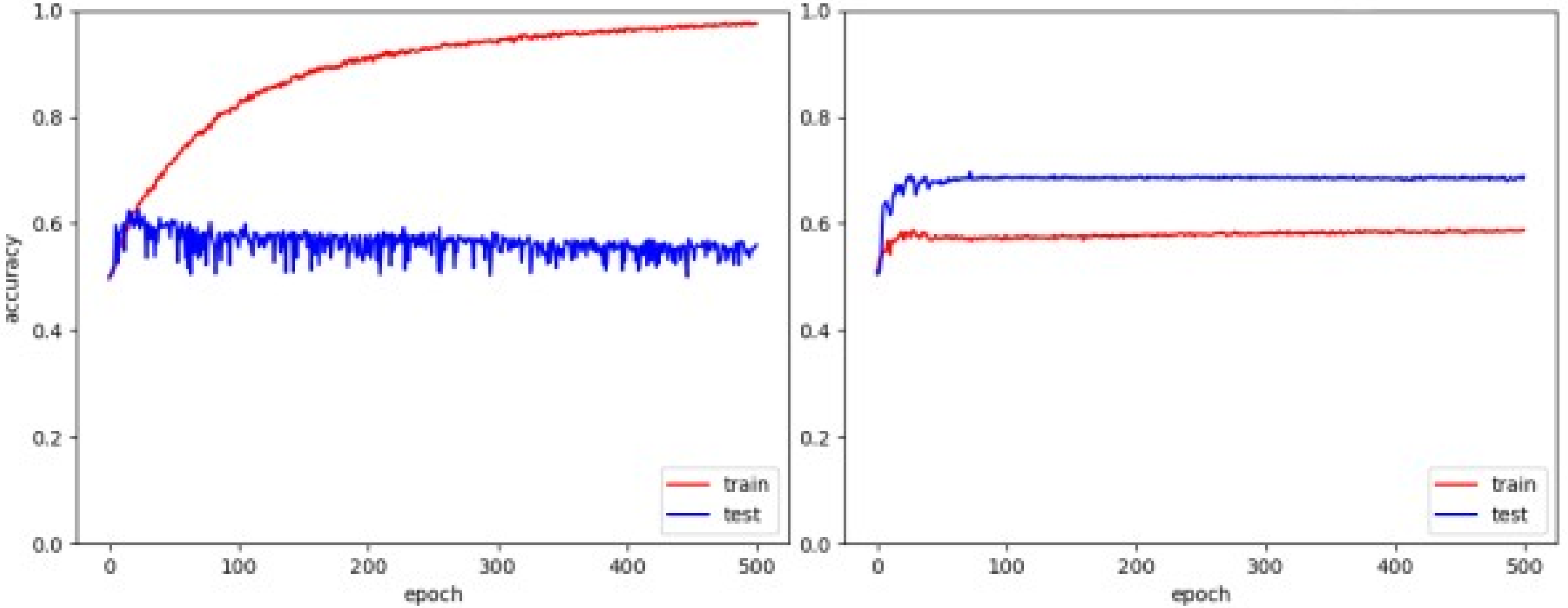
Nominal probability (1/N) $12.0 \cdot 10^{-4}$	Iteration number					
	i=1		i=2		i=49	
u_i-values	mislabeled images	correct labeled images	mislabeled images	correct labeled images	mislabeled images	correct labeled images
>>0	0	1	0	1	0	2
≈ 0	131	264	64	194	1,346	3,135
$-3.0 \cdot 10^{-5}$	0	0	24	63	2	1
$-6.0 \cdot 10^{-5}$	3,205	4,739	67	153	3	6
$-9.0 \cdot 10^{-5}$	0	0	3,181	4,593	0	0
$-12.0 \cdot 10^{-5}$	0	0	0	0	1,985	1,860
Total of images	3,336	5,004	3,336	5,004	3,336	5,004

Final test accuracy in AIRBUS for ERM and RRM (ADH-LP / SGD)

Method	Corrupted training data percentage				
	40%	30%	20%	10%	0%
ERM	0.560	0.629	0.681	0.735	0.769
RRM ($\mu = 0.5$)					
$\theta = 0.15$	0.603	0.739	0.745	0.774	0.764
$\theta = 0.20$	0.671	0.755	0.753	0.775	0.769
$\theta = 0.25$	0.687	0.733	0.757	0.769	0.767
$\theta = 0.30$	0.684	0.744	0.764	0.765	0.774
$\theta = 0.35$	0.661	0.747	0.764	0.770	0.779

The values highlighted in gray represent the cases where RRM outperform or match ERM.

Training and test accuracy for ERM (left) and RRM ADH-LP/θ = 0.25 (right) on AIRBUS with 40% of contamination



Evolution of u-vector across ADH-LP / SGD in AIRBUS (40% of contamination and θ = 0.25)

Nominal probability (1/N) $12.0 \cdot 10^{-5}$	Iteration number					
	i=1		i=2		i=49	
u_i-values	mislabeled images	correct labeled images	mislabeled images	correct labeled images	mislabeled images	correct labeled images
>>0	1	0	1	0	1	0
≈ 0	418	1,257	312	1,117	1,192	3,591
$-3.0 \cdot 10^{-5}$	0	0	142	590	7	1
$-6.0 \cdot 10^{-5}$	2,917	3,747	106	140	2	2
$-9.0 \cdot 10^{-5}$	0	0	2,775	3,157	5	6
$-12.0 \cdot 10^{-5}$	0	0	0	0	2,129	1,404
Total of images	3,336	5,004	3,336	5,004	3,336	5,004

Conclusions and Future Work

Despite the improvements in accuracy and robustness achieved through the RRM method, there is still room for improvements.

In the current setup, *u*-optimization excludes up to 64% of mislabeled images in all experiments where RRM outperforms ERM. Perhaps a greater percentage can be achieved by employing novel approaches in the *u*-optimization procedure.

Maritime surveillance capabilities can be significantly improved by integrating advanced models like the enhanced ADH-LP and ADH-SUB algorithms in automated surveillance systems, particularly those employing unmanned vehicles

Conclusions

In conclusion, achieving robustness in machine learning models is crucial for their real-world deployment and reliability. It requires careful consideration of various techniques, evaluation methodologies, and trade-offs.

RRM demonstrates its potential as a robust and resilient method for handling varying degrees of label corruption in both datasets. In all cases, the ADH-LP architecture under the SGD optimizer performed remarkably well, showing the ability to delay performance degradation even under high corruption levels, establishing it as the top choice across all scenarios

42nd International Symposium on Military Operational Research (42 ISMOR)

Robust Machine Learning for Naval Image Classification in the Blue Amazon

Gabriel Custódio Rangel [a], Luiz Frederico H. S. B.Teixeira [a], Victor Benicio A. S. Alves, Igor Pinheiro de A. Costa [a,], Johannes O. Royset [b], Eric C. Eckstrand [c]

[a] *Naval Systems Analysis Center (CASNAV), Rio de Janeiro, RJ, Brazil*

[b] *University of Southern California (USC), Los Angeles, CA, EUA*

[c] *Naval Postgraduate School (NPS), Monterey, CA, EUA*