Change Detection Algorithm with Hardware Constraints

Prof. Ingo Sander[1], Postdoc Marcello Costa[2]
October 1, 2024 — [1]KTH Royal Institute of Technology, [2]Cisb-Saab

Contents

Change Detection on Synchronous MoC → rapid data analysis:

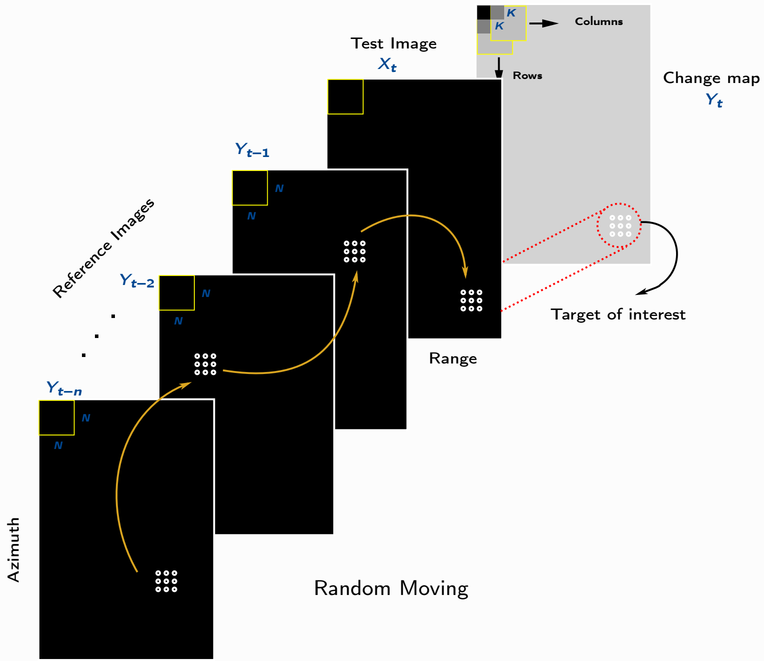
1. **Elementary Change Detection Model:** → Low complexity and power consumption, memory requirements, and parallelized structure[1]
2. **Time-series methods**[2] → ARIMA, PCA, SVM, Random Forest, NN, K-means, GLM, EM, SVD, XGboost, LSTM,..
3. **Temporal dataset: Ultra-wideband (UWB) SAR** → stable scatters in time, random motions of targets in the clutter as a first-order autoregressive AR(1) process

[1] Rabhi *et al.*, "Patterns and skeletons for parallel and distributed computing," Springer Science & Business Media, 2003.

[2] Wu, Suya, et al. "Quickest Change Detection for Unnormalized Statistical Models," *IEEE Trans. on Inf. Theory*, v. 70, n. 2, pp. 1220-1232, 2024.

Problem Introduction and Background

Time series Prediction: $$\hat{Y}_t = f(X_t, Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}) + \epsilon_t, \tag{1}$$



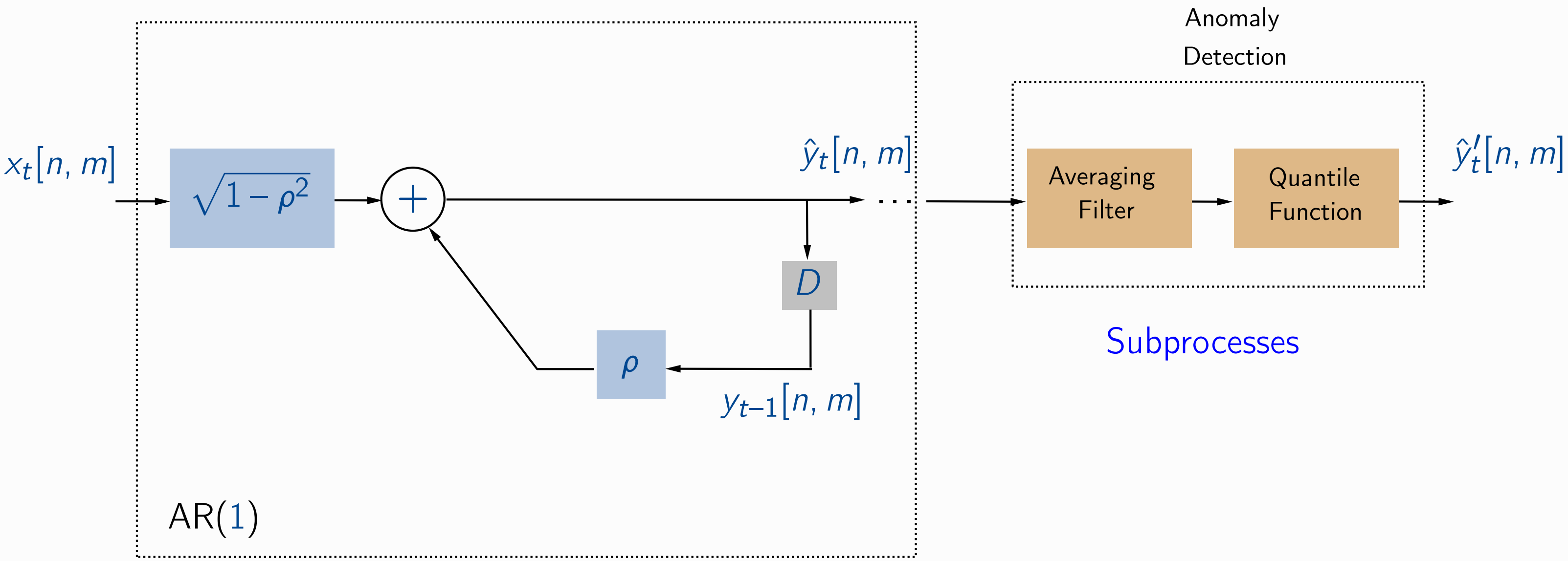
Combinational solution: 1-lag

X_t	Y_{t-1}	Change
0	0	No change
0	1	No change
1	0	Change
1	1	No change

Note: Anomaly = 1, background = 0.

Change Detection Algorithm: 2D-AR(n) on Synchronous MoC

Time-Spatial Implementation:



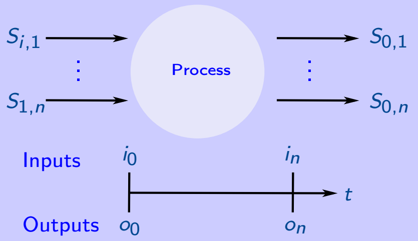
Synchronous Model of Computation

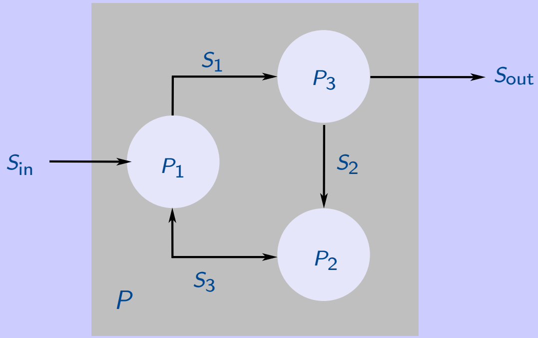
Subprocesses

Change Detection Algorithm: 2D-AR(n) on Synchronous MoC

Synchronous MoC[5]**:** Synchronized parallel components, running in successive computation steps, where all components perform some quantum of computation.

1. Output is synchronous with input
2. Internal actions are instantaneous





[5]Albert Benveniste and Gérard Berry. "The synchronous approach to reactive and real-time systems," *Proceedings of the IEEE*, 79(9):1270–1282, September 1991.

Change Detection Algorithm: 2D-AR(n) on Synchronous MoC

Change Detection with AR Model

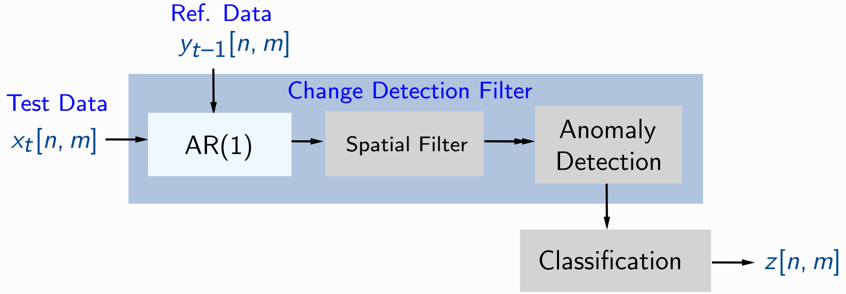
▶ForSyDe Framework[5]/Haskell

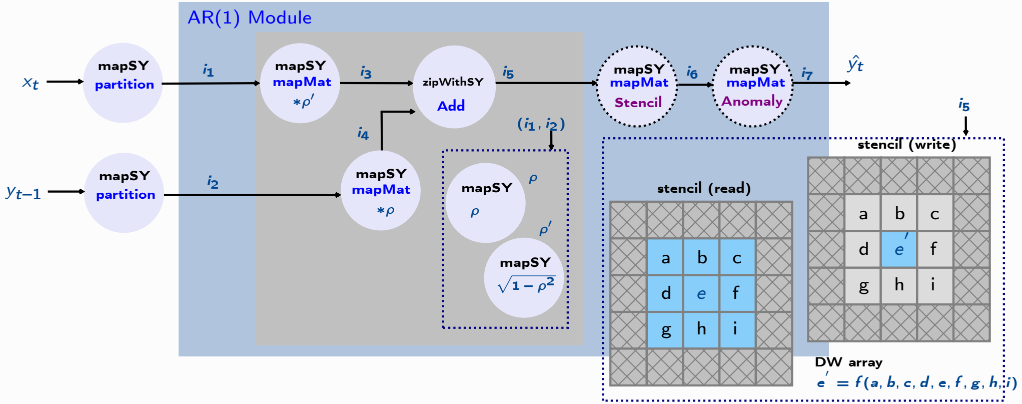
Synchronous MoC constructors:

- Combinational → `zipWithSY`, `mapSY`
- Delay → `delaySY`
- Sequential → `mooreSY`, `mealySY`

Parallel stencil computation

- Data.Massiv.ArrayFramework[6]



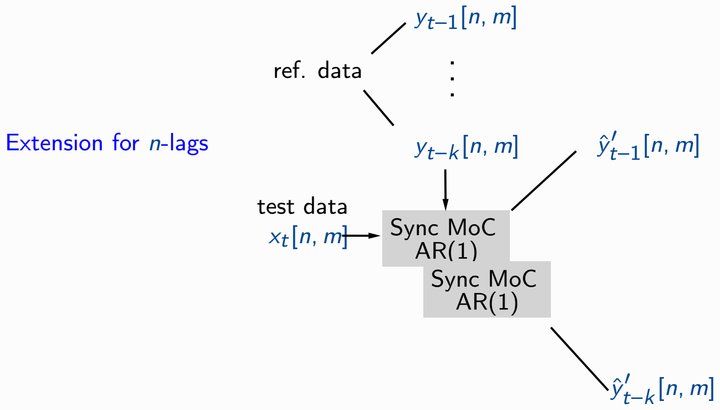


[5] https://forsyde.github.io/

[6] https://hackage.haskell.org/package/massiv

Extension: n-lags Model

- Hardware efforts × detection performance are balanced over high-parallelized Sync MoC structure by $[N, k]$ selection, enabling fast data analysis.
- Extension for a Markov chain in multitemporal (n-lags) improves the detection performance (considering relevant statistical dependencies), replicating the elementary MoC structure.

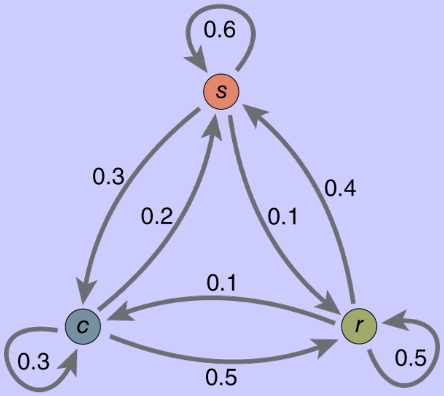


n-lags Model

Markov-chain: all states are observable and probabilities converge over time (dependencies are included)

$$r_{i,j}[n] = \sum_k r_{i,k}[n-1] P_{k,j}$$

$$\begin{cases} r_{i,j}[n] \rightarrow & \text{Prediction going to state } i \text{ to } j \\ r_{i,k}[n-1] \rightarrow & \text{recursion from state } i \text{ to } k \\ P_{k,j} \rightarrow & \text{Transition probability from state } k \text{ to } j \end{cases}$$



where $r_{i,j}[n] \Rightarrow \hat{y}'_t$ for each incremental lag.

Proposition: n-lags 2D-AR Model on Markov-Chain.

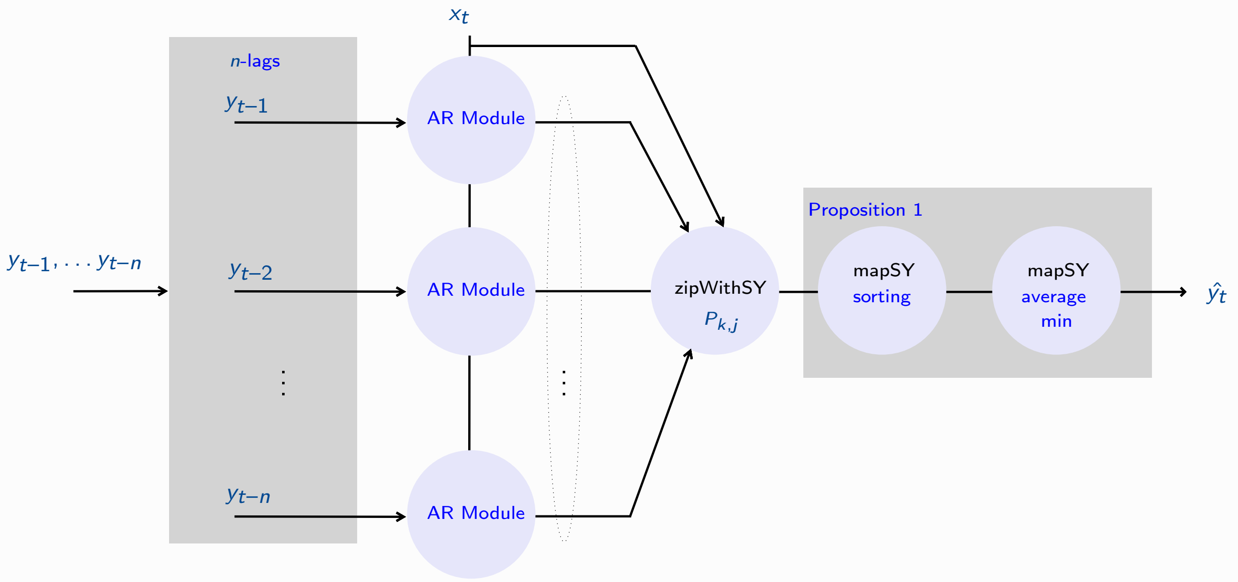
Let $r_{i,j}[n]$ be the prediction extracted from a Markov chain. For change detection, Let $\tilde{r}^{\ell}_{i,j}[n]$ denote the set of interest involving AR(1) subject to the lowest probability. Then we will the unique sequence

$$\tilde{r}^{\ell}_{i,j}[n] = \bigcup_{i=0}^{\ell-1} \min\{r_{i,j}[n] \backslash \tilde{r}^{\ell}_{i,j}[n]\}, \tag{2}$$

which represents the most likely available lags to detect changes anomalies in $r_{i,j}[n]$ with minimum false alarms.

□

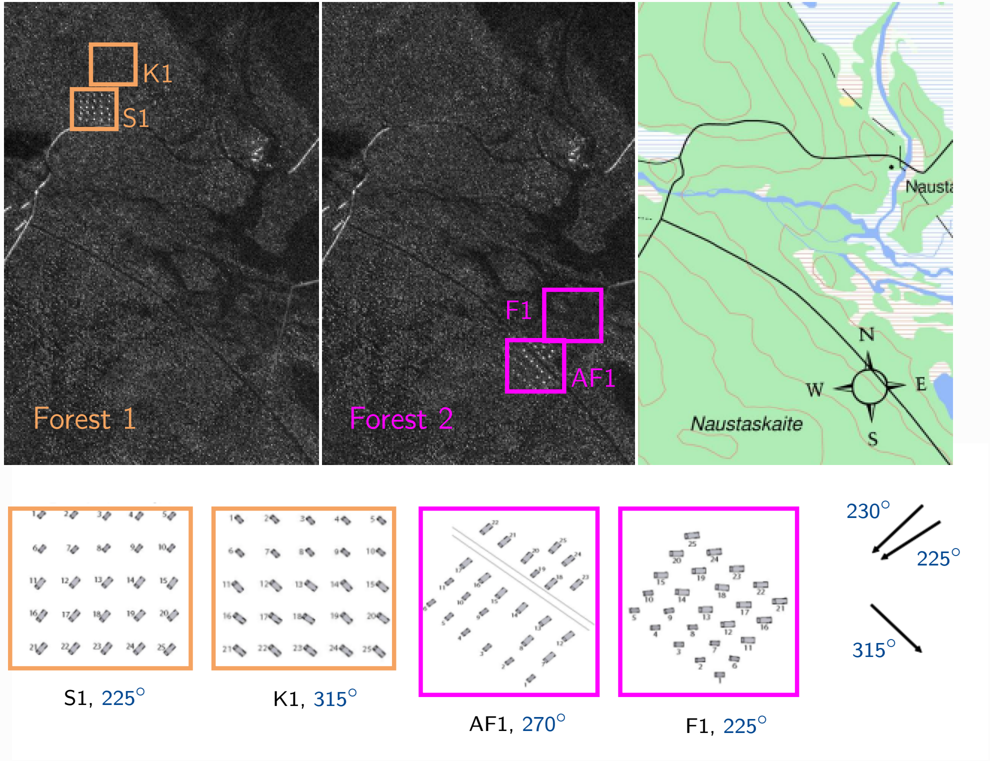
Synchronous MoC on ForSyDe



CASE I: Change Detection on CARABAS-II

CARABAS-II dataset[7]:

- 24 images: magnitude VHF ($20 - 90$ MHz) SAR HH-polarized with 1 m resolution (2×3 km^2)
- 4 Targets position: S1, K1, F1, and AF1
- 6 passes: 3 Flight directions under ground RFI sources ON/OFF
- Application: Through-foliage detection



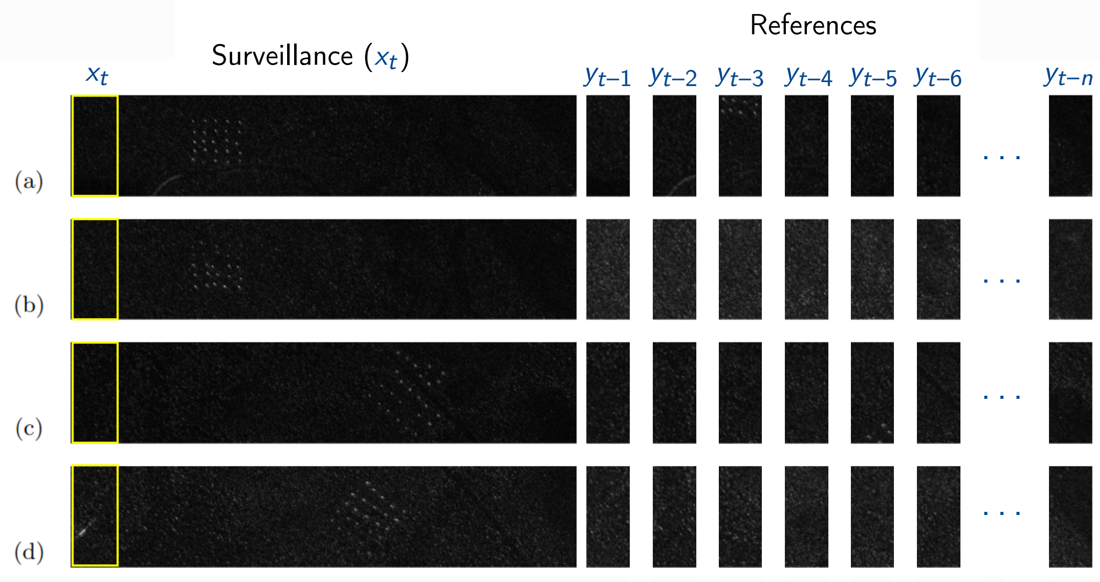
[7] Ulander, L.M, et al. "Change detection for low-frequency sar ground surveillance," *IEE Proceedings-Radar, Sonar and Navigation* , 152(6), pp. 413–420, 2005.

CASE I: Change Detection on CARABAS-II

Exp. No.	Surveillance image Mission	Surveillance image Pass	Reference image Mission	Reference image Pass	Known Targets	Area [km^2]
1	2	1	3	1	25	6
2	3	1	4	1	25	6
3	4	1	5	1	25	6
4	5	1	2	1	25	6
5	2	2	4	2	25	6
6	3	2	5	2	25	6
7	4	2	2	2	25	6
8	5	2	3	2	25	6
9	2	3	5	3	25	6
10	3	3	2	3	25	6
11	4	3	3	3	25	6
12	5	3	4	3	25	6
13	2	4	3	4	25	6
14	3	4	4	4	25	6
15	4	4	5	4	25	6
16	5	4	2	4	25	6
17	2	5	4	5	25	6
18	3	5	5	5	25	6
19	4	5	2	5	25	6
20	5	5	3	5	25	6
21	2	6	5	6	25	6
22	3	6	2	6	25	6
23	4	6	3	6	25	6
24	5	6	4	6	25	6
Total					600	144

Include all lags y(t-n) to improve the prediction model

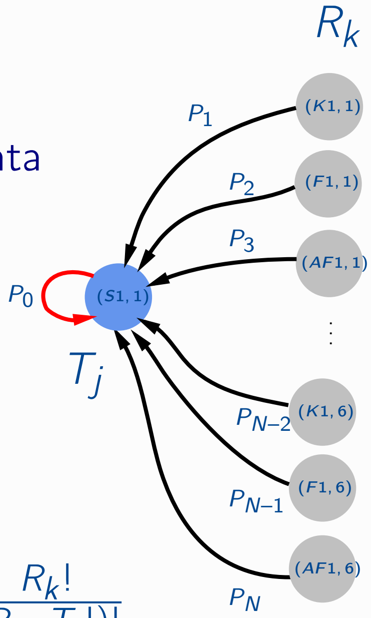
Campaign sample data: 2×0.5 km (15%)



CASE I: Change Detection on CARABAS-II

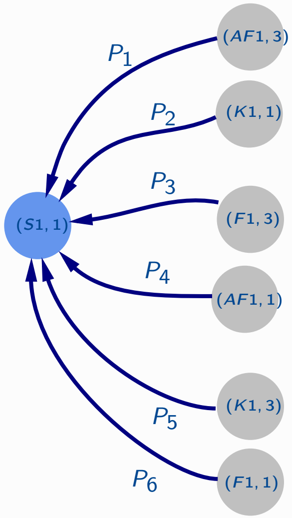
R: reference data

T: test data

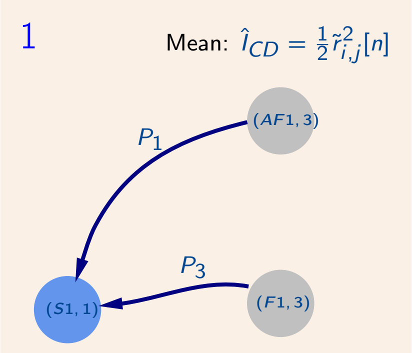


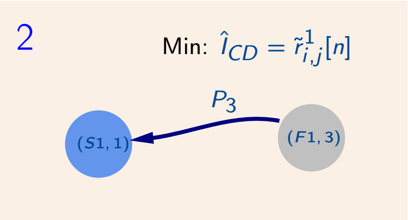
$$P_{k,j} = \prod_{j=1}^{J} \prod_{k=1}^{K} \frac{R_k!}{(R_k - T_j!)!}$$

Best pairs



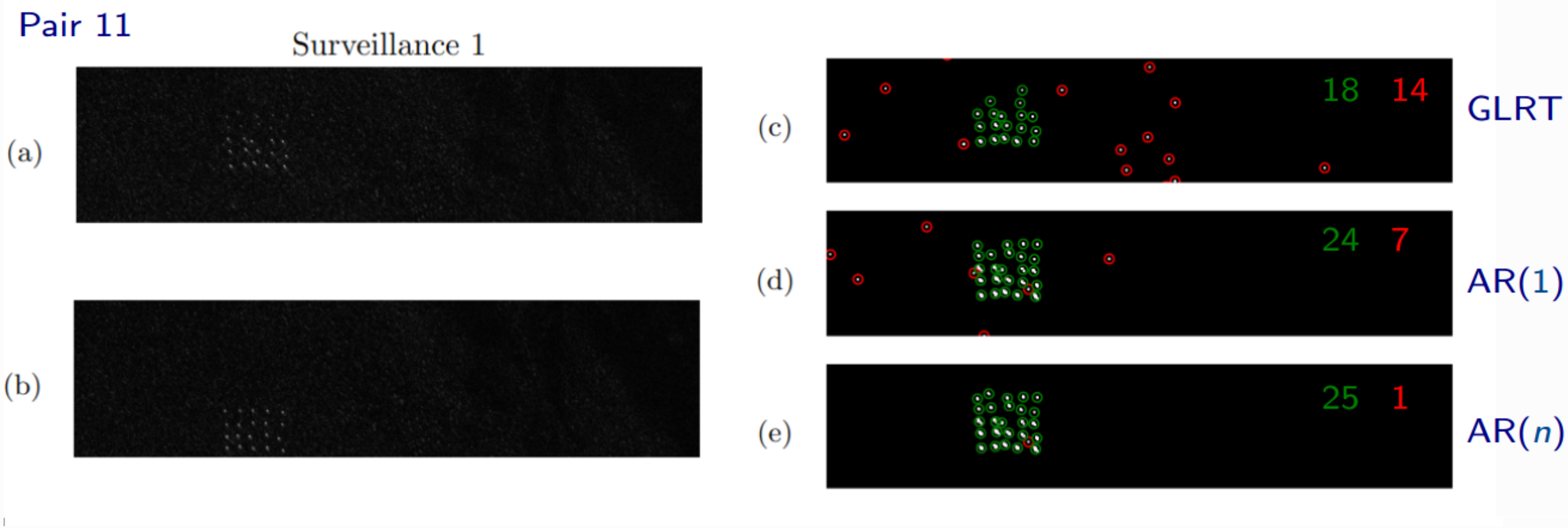
Decision Criteria





CASE I: Change Detection on CARABAS-II

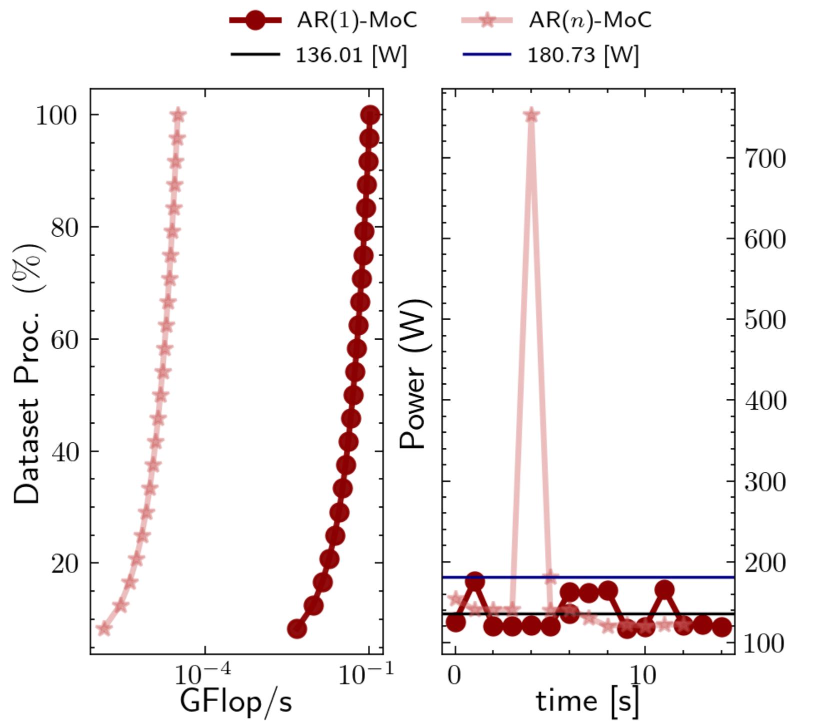
$2 \times 0.5\ \mathrm{km}^2$ (15%); $N = 40$; AR(n), $n = [1, 6]$



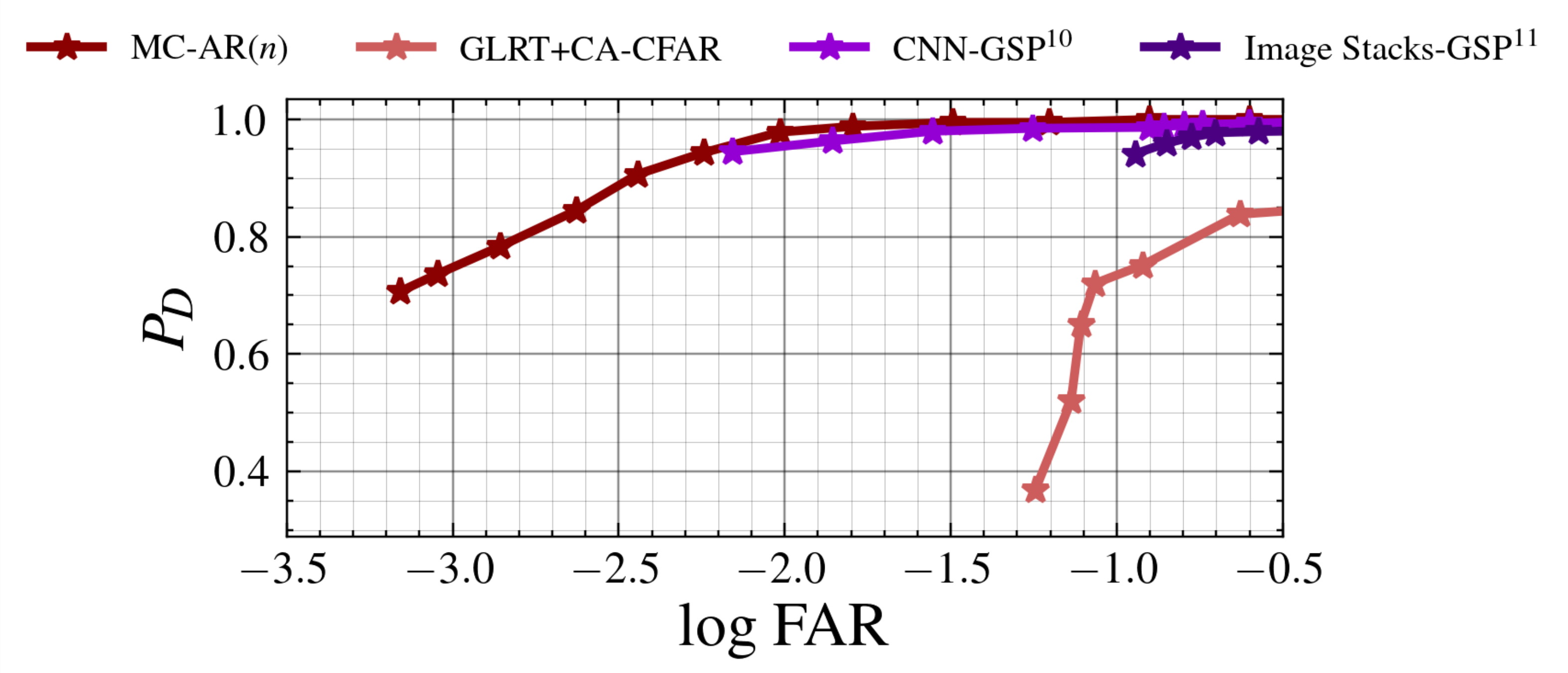
Energy-Complexity Performance

	AR(1)		AR(n)/6	
Instance	GLRT	CFAR	sf	q-ECDF
Complexity	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2K^2)$	$\mathcal{O}(N^2k^2)$	$\mathcal{O}(N^2 \log N^2)$
Runtime (sec)	8.47	242.60	10.65	7.00

Note: sf = spatial filtering stage with averaging filter.



CASE I: Change Detection on CARABAS-II



[10] Vinholi, J.G., et al., 'Change Detection Based on Convolutional Neural Networks Using Stacks of Wavelength-Resolution Synthetic Aperture Radar Images," in IEEE TGRS, vol. 60, pp. 1-14, 2022.

[11] Palm, B.G., et al., "Wavelength-Resolution SAR Ground Scene Prediction Based on Image Stack," Sensors 2020, 20(7).

Conclusions

- The generalized n-lags model has potential detection improvement over high-parallelized structure.
- Embedded SW application for heterogeneous architectures: CPUs, GPU or FPGA.
- Notes:

$$\tilde{r}^{\ell}_{i,j}[n] = \bigcup_{i=0}^{\ell-1} \min\{r_{i,j}[n] \backslash \tilde{r}^{\ell}_{i,j}[n]\} \rightarrow \text{Generalized model,}$$

$$\begin{cases} \ell = 1 \Rightarrow \text{best bi-temporal AR(1)} \\ \ell | \min PFA \Rightarrow \ell < k \end{cases}$$

Next Steps

1. Target architectures

2. Parallel scheduling

3. Test performance

KTH
VETENSKAP
OCH KONST