Continual Learning for on-ground satellite health monitoring

Gabriele De Canio, Rudy Semola*, Valerio De Caro*

European Space Operations Centre, European Space Agency

* Department of Computer Science, University of Pisa

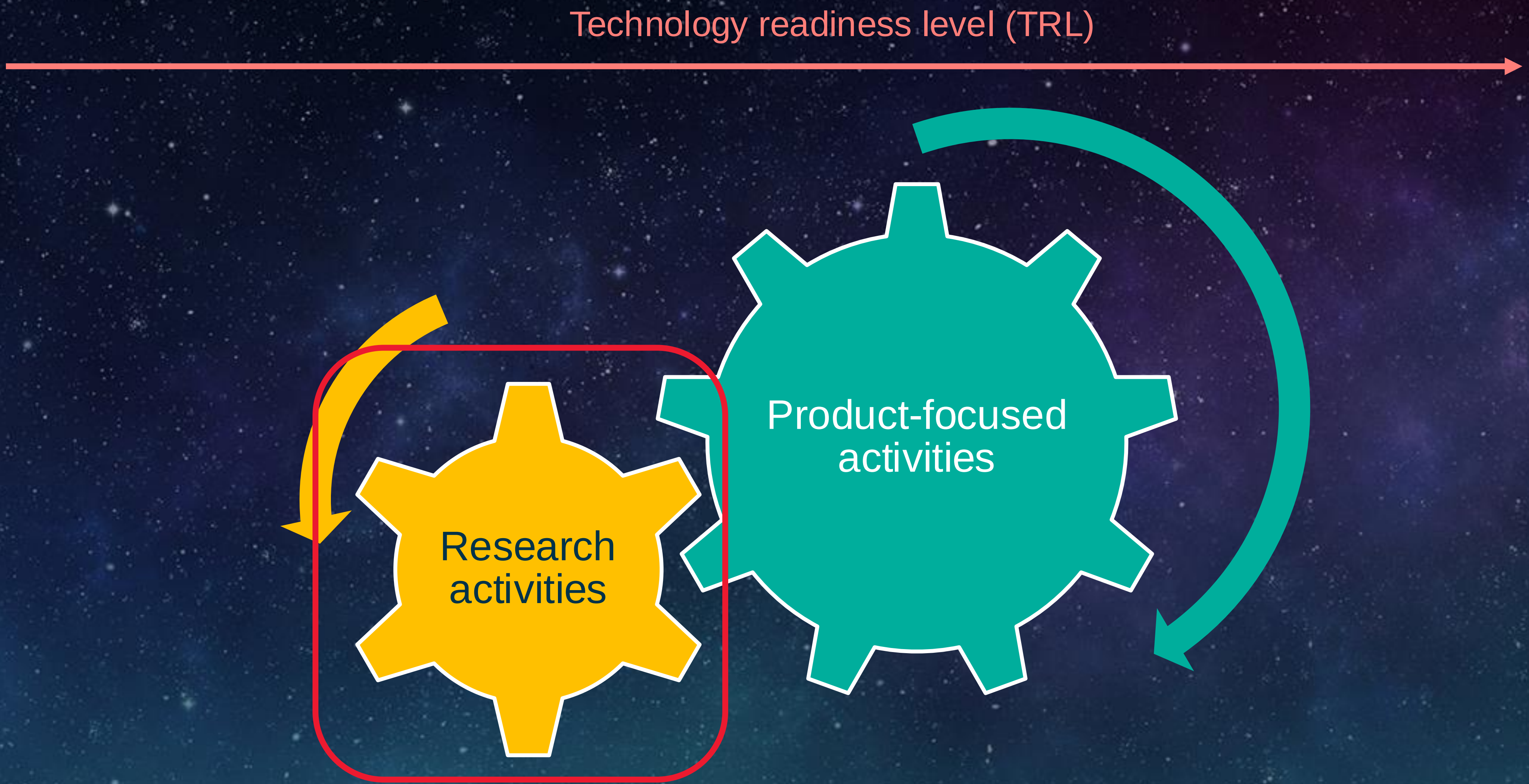
AI and Data Foundation Roadmap

5 capabilities under development

Data Handling, Management, Governance, Assurance

- **AI-enhanced modelling and simulation**
- **AI-automated content generation and AI-enhanced user interaction**
- **Automated health monitoring and control**
- **AI-enhanced decision recommendation and planning**

Spinning in innovation



Satellite health monitoring

Challenges for AI and automation

High Dimensionality and Large Volume

- Multivariate time series with numerous telemetry parameters and telecommands
- Complex dependencies between telemetry parameters and telecommands

Changing Spacecraft Modes and Environments

- Visible shifts in data distribution
- Altered interactions between telemetry parameters and telecommands

Measurement errors

- Data gaps
- Outliers

Applications of AI in satellite health monitoring

Despite challenges, AI is demonstrating its benefits in automating space operations

SMAI

Spacecraft Monitoring AI-powered suite

- Trend analyzer
- 4caster
- Root-cause assistant
- Anomaly detector

Continual Learning (CL)

Definition	The ability of a model to learn continuously from a stream of data
Efficiency	Ensures constant computational and memory resources
Incremental Development	Allows for the model to evolve as input data evolves over time
Applications	Powerful in scenarios where data distribution shifts over time and we need fast adaptation

Continual Learning (CL)

Definition	The ability of a model to learn continuously from a stream of data
Efficiency	Ensures constant computational and memory resources
Incremental Development	Allows for the model to evolve as input data evolves over time
Applications	Powerful in scenarios where data distribution shifts over time and we need fast adaptation

Objectives

Demonstrate usage of CL as a new paradigm to reduce computational cost without compromising quality of ML model

Usage of the publicly available ESA Anomaly Dataset ESA-ADB

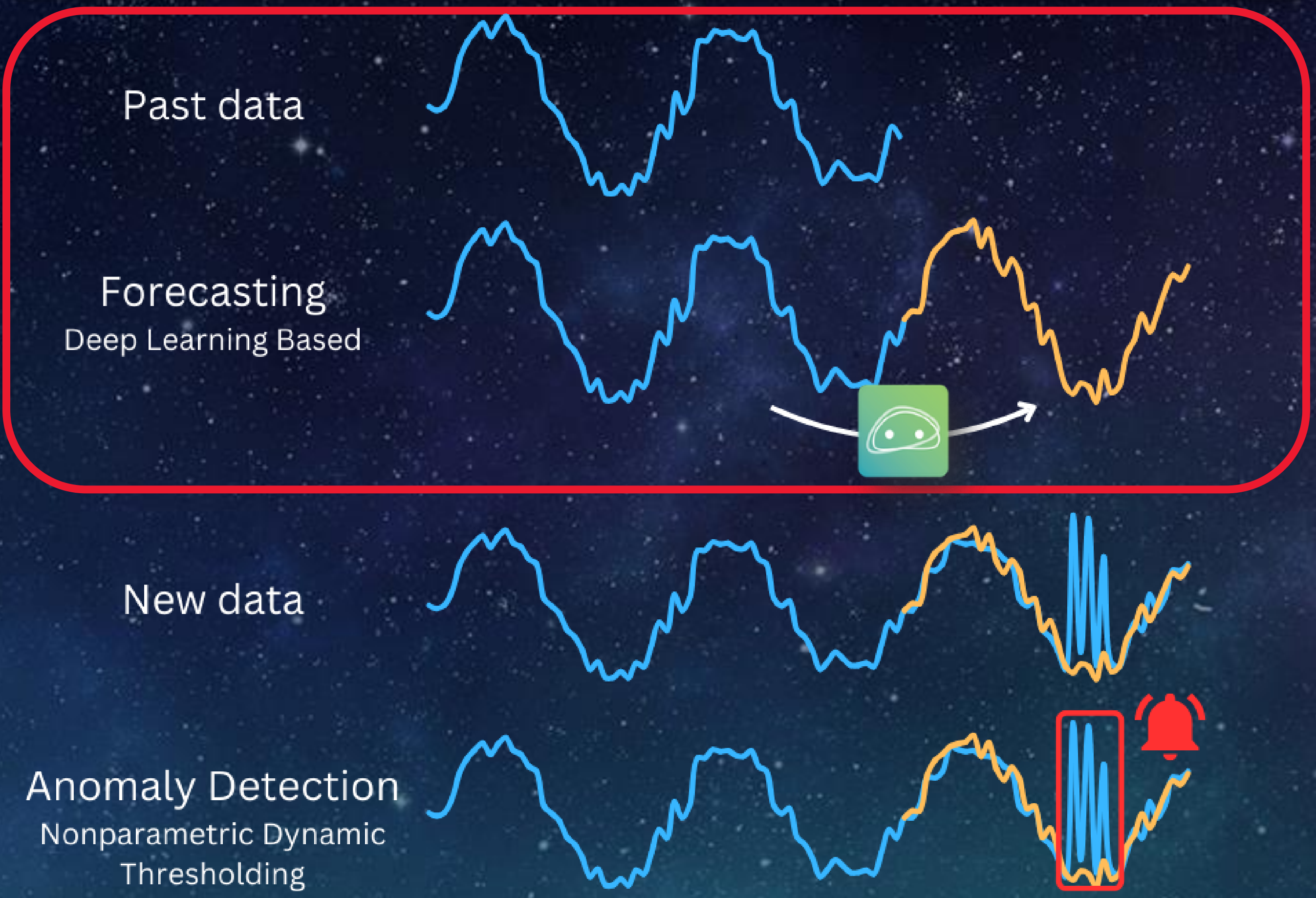
Focused on time-series forecasting (only 1 mission)

Estimate time window to produce reliable forecasts

Methodology

esa

Focus on Multi-Time series data forecasting for AD



Step 1 - time-series forecasting

- Model
  - For groups: LSTM (2 layers, 80 units)
  - For subsystem or target channels: LSTM (2 layers, 134 units)

Approaches

- Cumulative
- Naïve (CL)
- Replay (CL)

Metrics

- Mean Square Error (performance)
- Memory and Time (compute)

Methodology

Cumulative

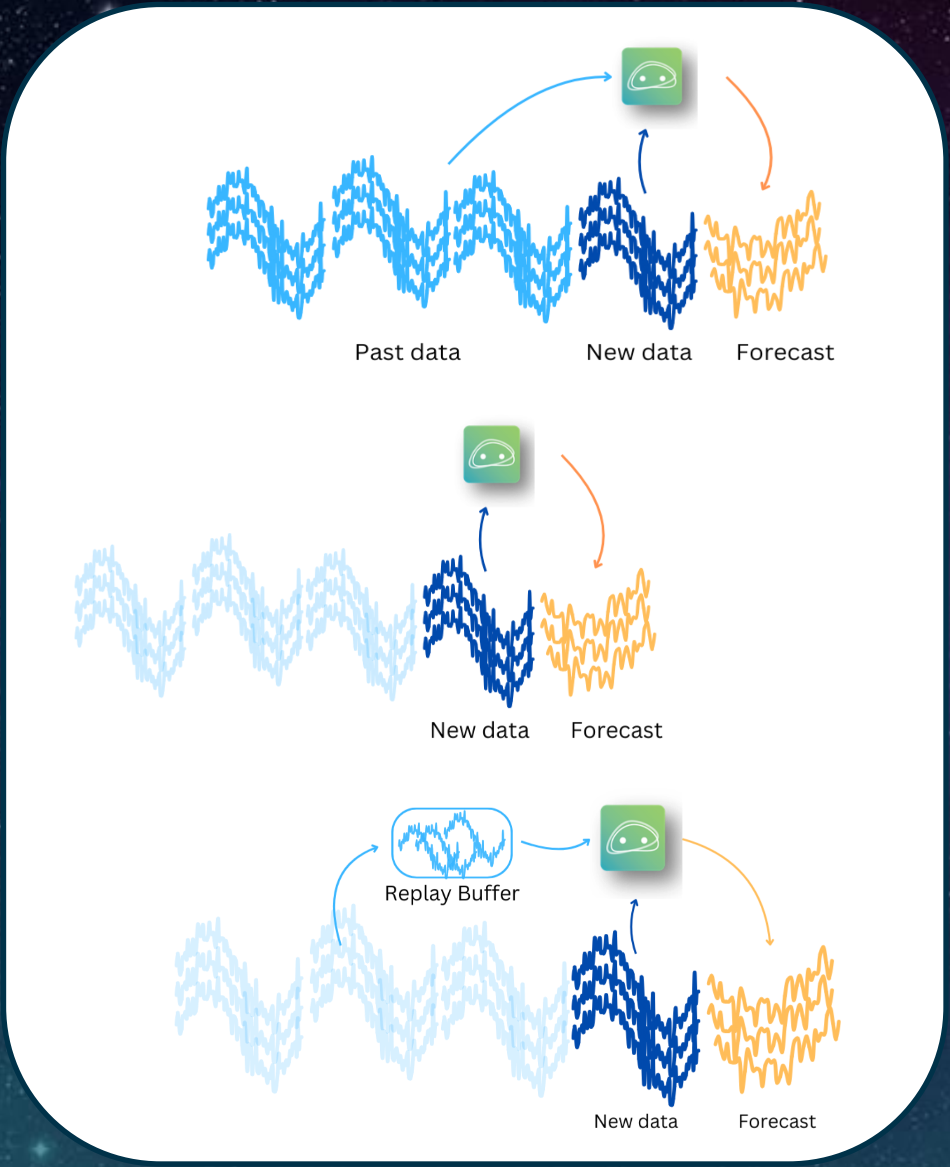
- Advantages: Excellent long-term generalization and robustness.
- Disadvantages: Computationally expensive and resource-intensive.

Continual Naïve

- Advantages: Quick Adaptation and computationally efficient.
- Disadvantages: Limited generalization capacity for long-term trends.

Continual Replay

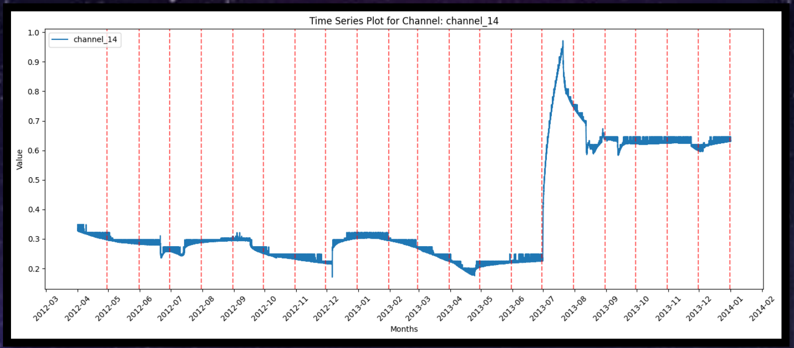
- Advantages: Balances computational efficiency and predictive accuracy.
- Disadvantages: Performance heavily depends on the selection of the buffer size.

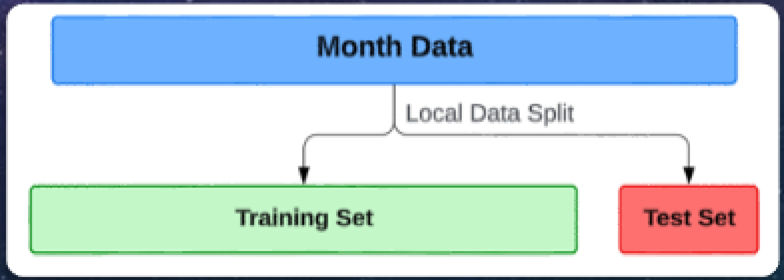


Setup Validation

Dataset: ESA-ADB Mission 1 (anonymized)

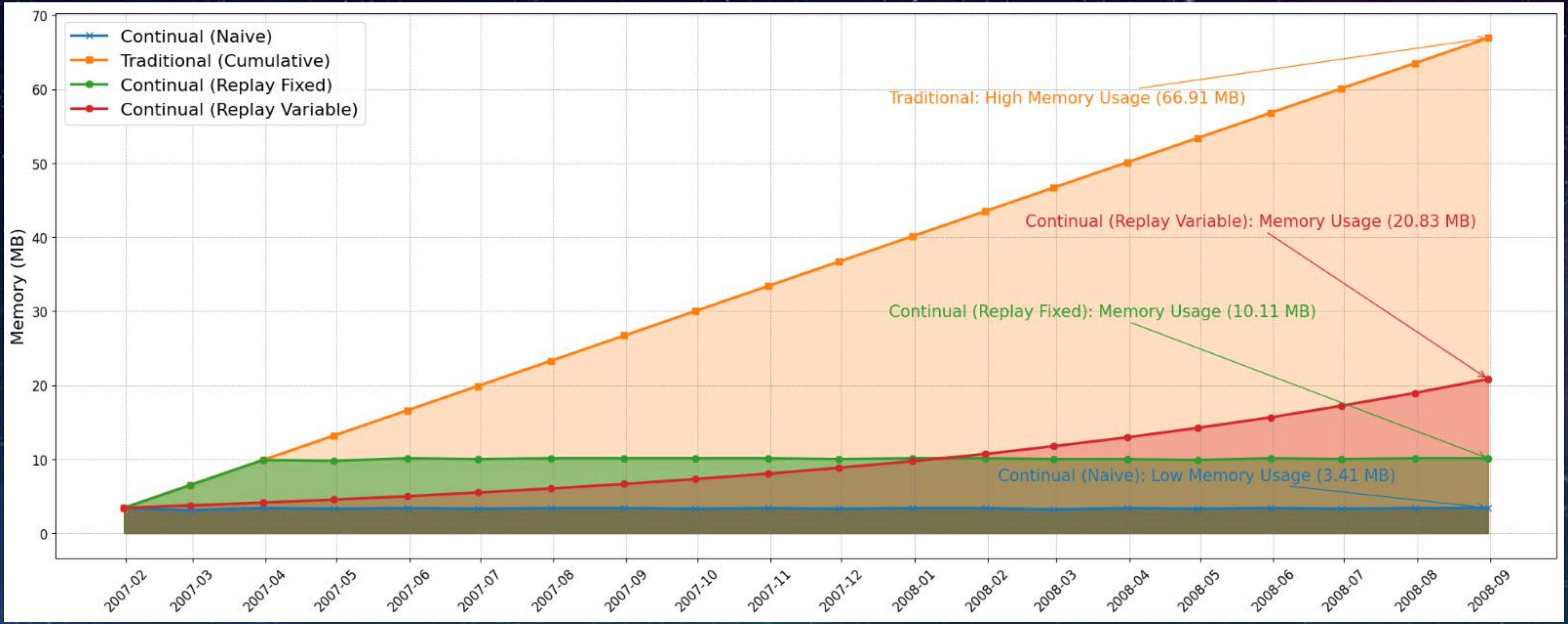
- Focus on channels with **drift and complex behaviors**:
  - Groups 4, 7, 13 and Subsystem 6
- Data splitting (Experience): 1 month
- Evaluation (Agile Machine Learning principle)
  - Incremental development, simulating real operational scenarios
  - Iterative approach, update every month





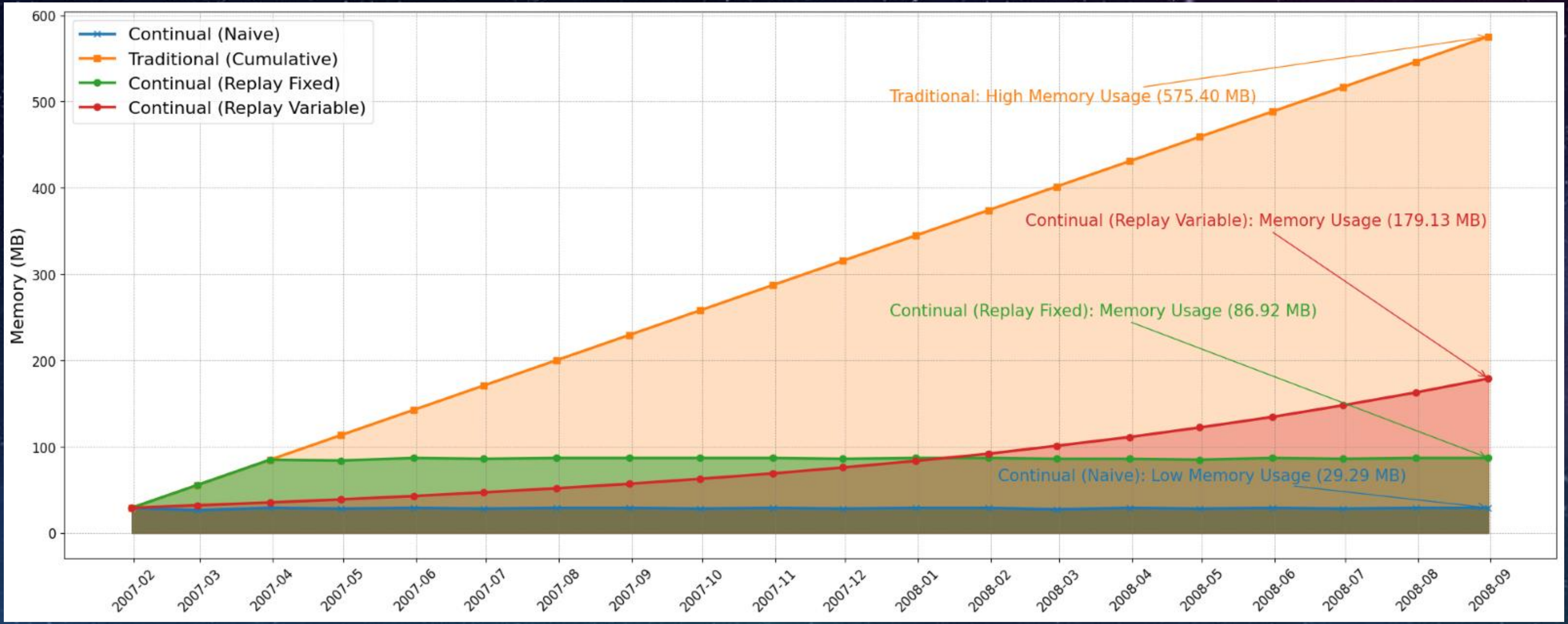
Results – efficiency

Group 13 (5 channels)



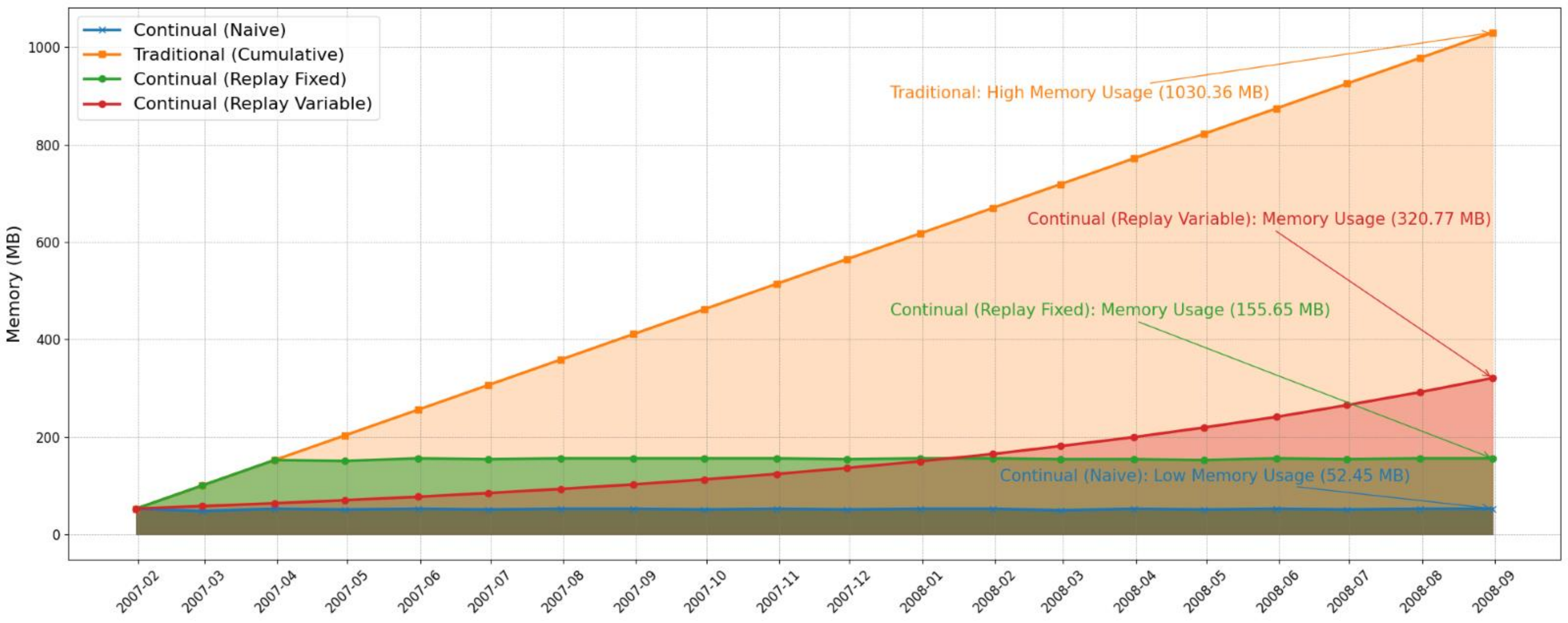
Results – efficiency

Subsystem 6 (42 channels)



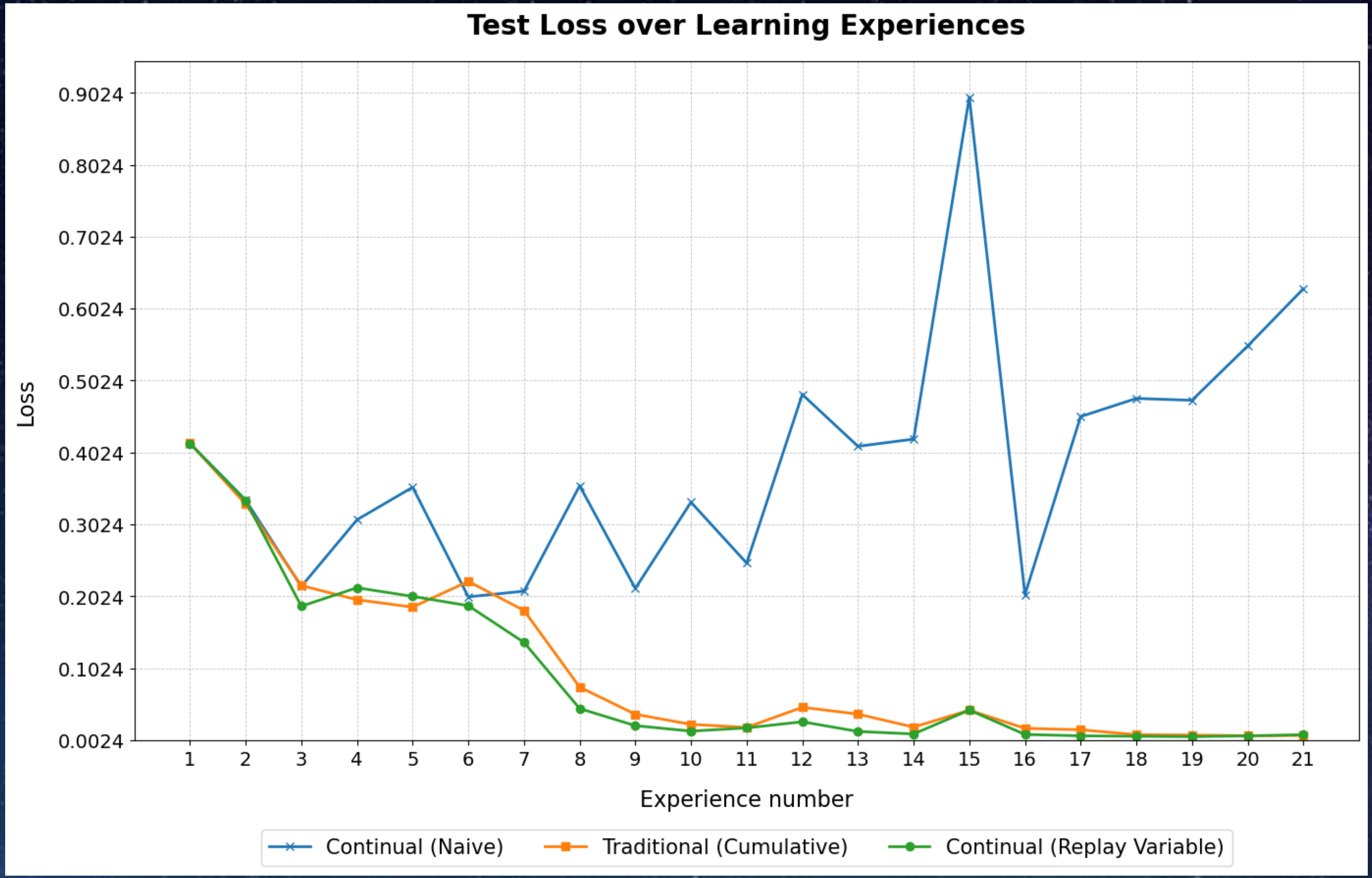
Results – efficiency

Target channels (56 channels)



Results – Performance

Group 13 (5 channels)

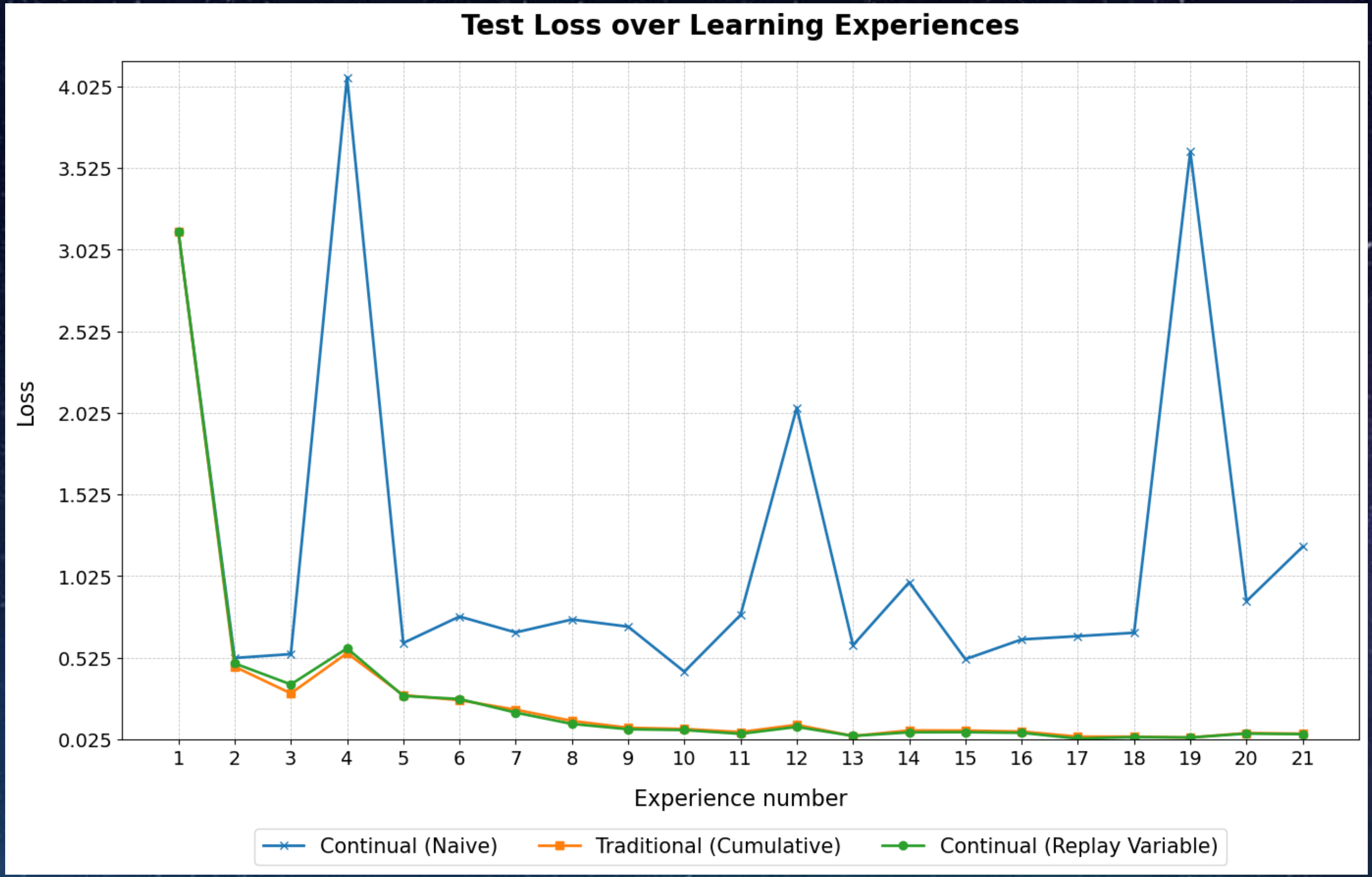


Average Loss (MSE) over experiences

Method	Average Loss (MSE)
Cumulative	0.1018
Naïve	0.3901
Replay (Variable)	0.0920

Results – Performance

Subsystem 6 (42 channels)

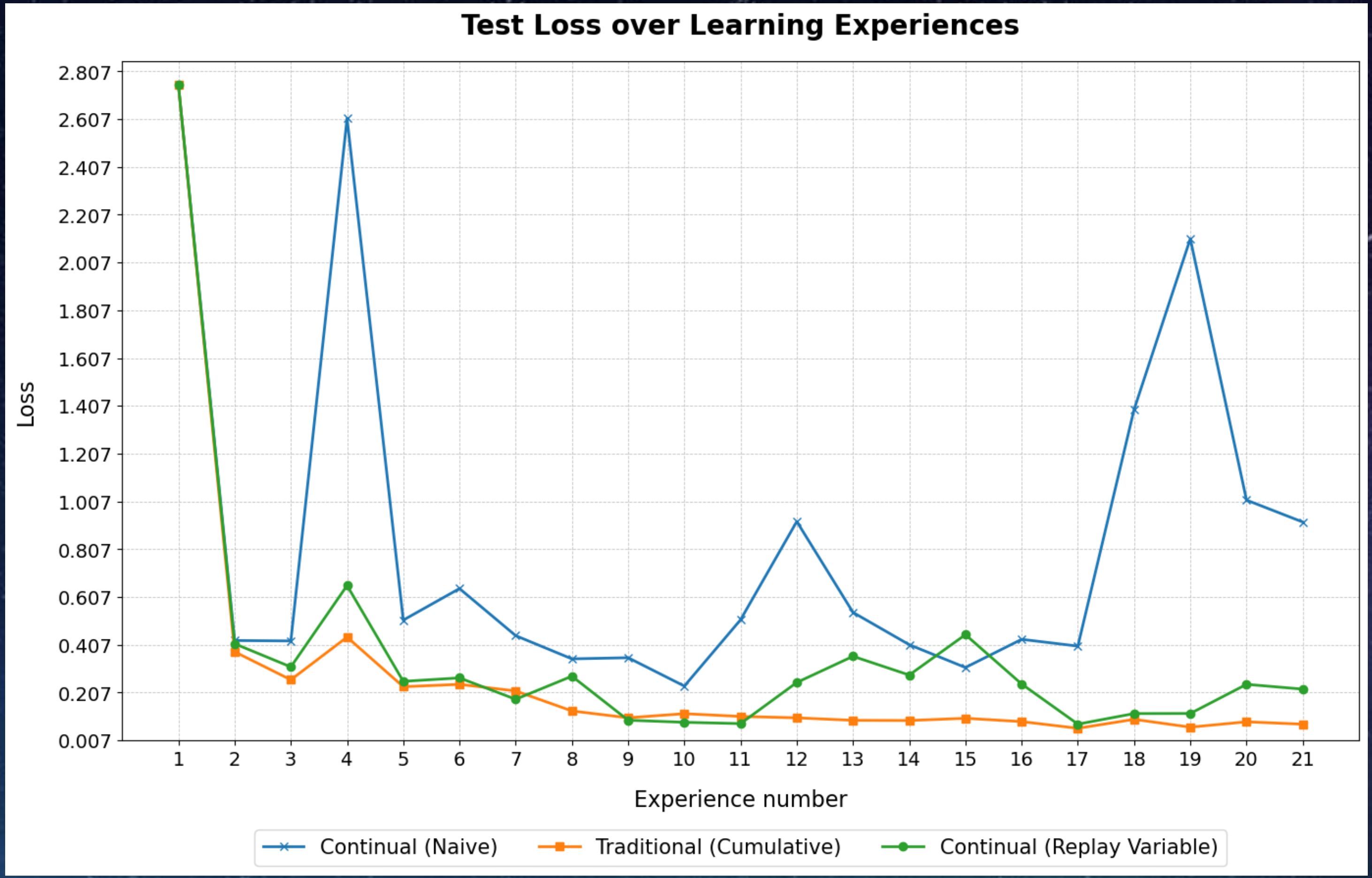


Average Loss (MSE) over experiences

Method	Average Loss (MSE)
Cumulative	0.2981
Naïve	1.1852
Replay (Variable)	0.2972

Results – Performance

Target channels (56 channels)



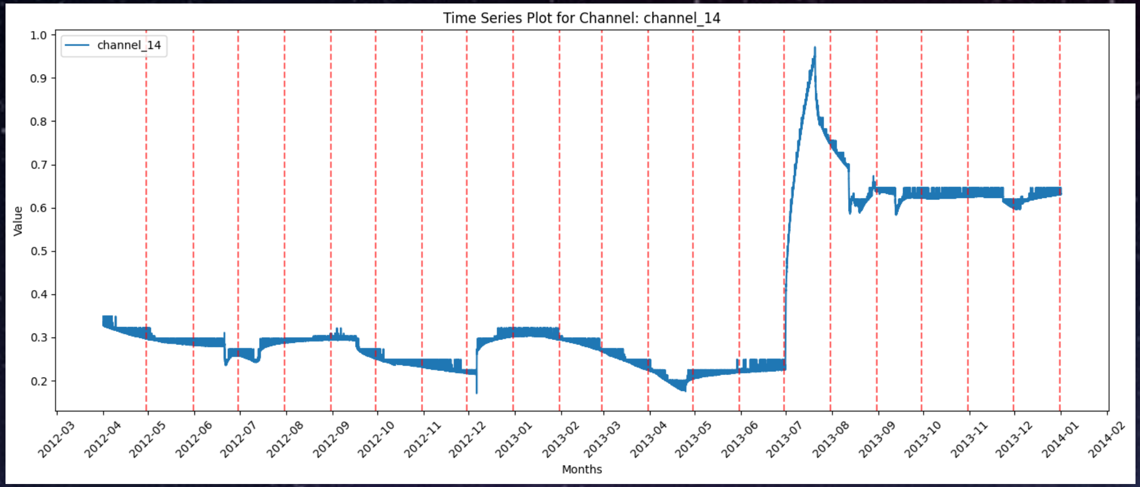
Average Loss (MSE) over experiences

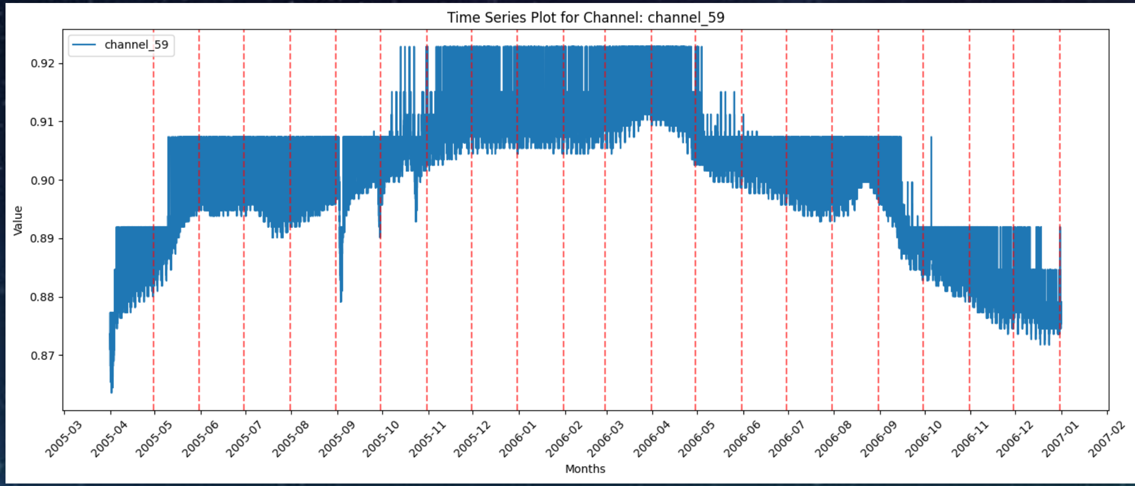
Method	Average Loss (MSE)
Cumulative	0.2766
Naïve	0.8427
Replay (Variable)	0.3276

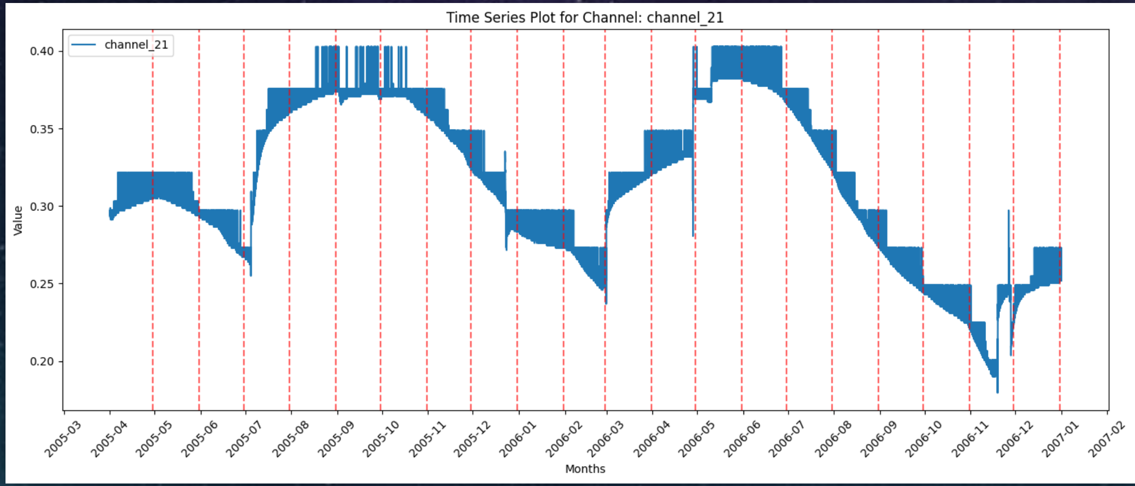
Telemetry Multi-Time Series Data – Challenges

Mission 1

- Several anomalies are hard to spot
- Several huge outliers
- Low signal-to-noise ratio
- Monotonically non-decreasing signals
- Include a severe ***concept drift*** in channels from groups 4, 7, and 13 (and Subsystem 6)
- There is a **visible seasonality with a long period length** (up to 6 months)
- Overabundance of telecommands







Conclusion

- ❑ Continual Learning is a promising ML paradigm for satellite health monitoring
- ❑ On ESA-ADB, Continual Replay (Variable) achieves the optimal balance between efficiency and efficacy
    - Subsystem 6: Up to 4x loss reduction (Naïve) and ~2x (Cumulative)
    - Subsystem 6: up to a 72% compute reduction (Cumulative)
- ❑ Explanation: buffer memory of (most informative) historical data enhances performance by improving both short-term adaptability and long-term generalization, while balancing efficiency, time, and memory requirement.

Conclusion

- **Future Direction**

  - Advanced Replay techniques to enhance forecasting capabilities and speed up deep learning-based anomaly detectors

  - Onboard use case

- Provide *Continual Learning as a Service* across ground segment applications to enable effective and agile machine learning practices.

Thank you!

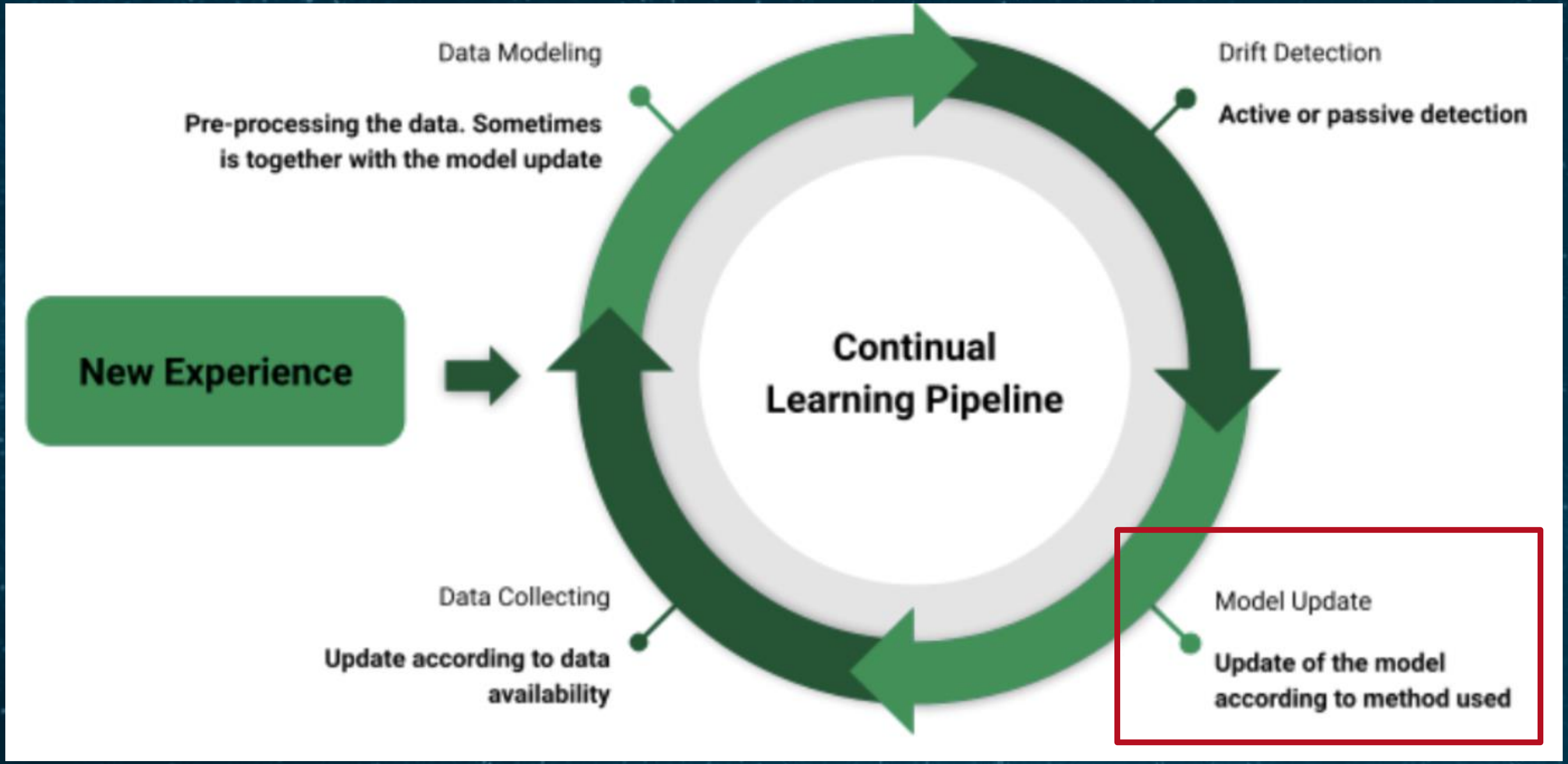
New Anomaly Detection Challenge on Kaggle using ESA-ADB. **Launch event at ESOC on 13 March**

Talks at GSAW 2025

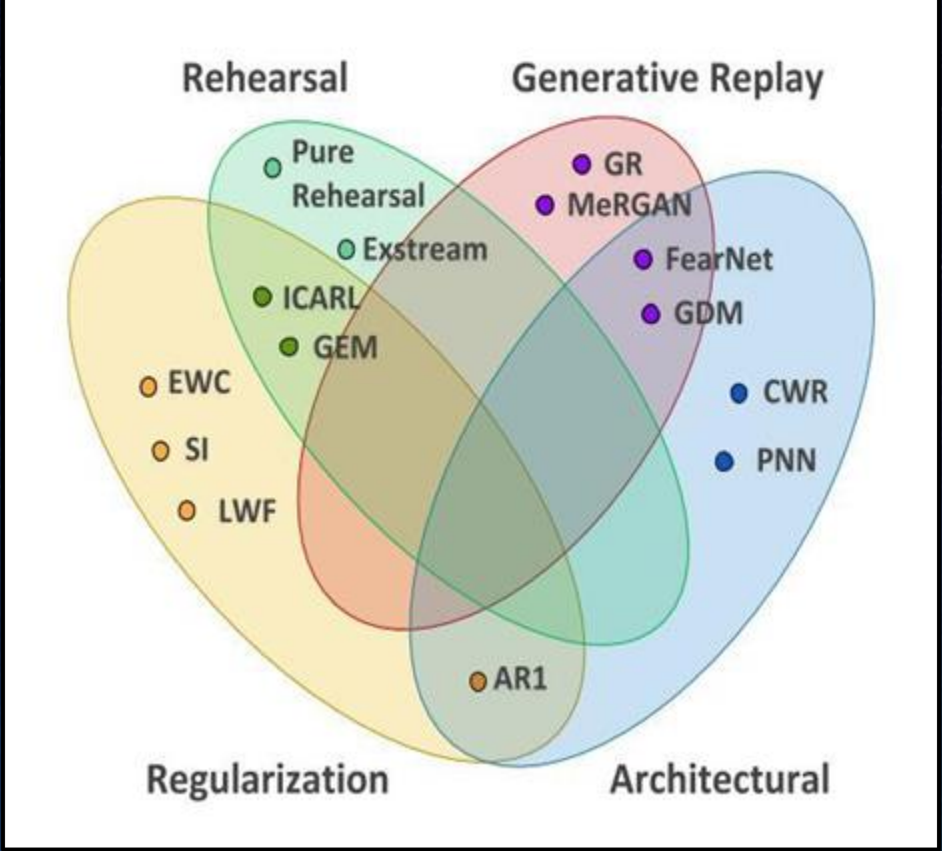
- Artificial Intelligence for mission operations automation roadmap
- Continual Learning for on-ground satellite health monitoring
- Enhancing Spacecraft Operations with Digital Twins Solutions
- Data-X: Pioneering the Future of Data in Mission Operations

Backup Slides

Continual Learning



Continual Learning Methodologies Landspace



Algorithm 1 Pseudocode explaining how the external memory RM is populated across the training batches. Note that the amount h of patterns to add progressively decreases to maintain a nearly balanced contribution from the different training batches, but no constraints are enforced to achieve a class-balancing.

$RM = \varnothing$
RM_{size} = number of patterns to be stored in RM
for each training batch B_i:
    train the model on shuffled $B_i \cup RM$
    $h = \dfrac{RM_{size}}{i}$
    R_{add} = random sampling h patterns from B_i
    $R_{replace} = \begin{cases} \varnothing & \text{if } i == 1 \\ \text{random sample } h \text{ patterns from } RM & \text{otherwise} \end{cases}$
    $RM = (RM - R_{replace}) \cup R_{add}$
