An Agent-Orchestrated Anomaly Detection Pipeline for Network Performance

Luigi Rachiele[1,2], Vincenzo Mancuso[1,3], Juan Marcos Ramírez[1]
IMDEA Networks[1], UC3M[2], University of Palermo[3]

Abstract

*Today's telecom **anomaly detection** relies on hand-engineered pipelines that break whenever data or scenario shifts. We propose a reasoning **agent** that **builds**, **tunes**, and **explains** its own detection pipeline by orchestrating tools in an adaptive loop, thereby turning a brittle process into an **auditable** one and laying the foundation for **agentic reasoning** on cellular networks.*

Background

- Modern cellular networks use machine learning to monitor performance and detect faults [1];
- These models are accurate but work as black boxes [2]: they tell operators that something is wrong, not **why.**

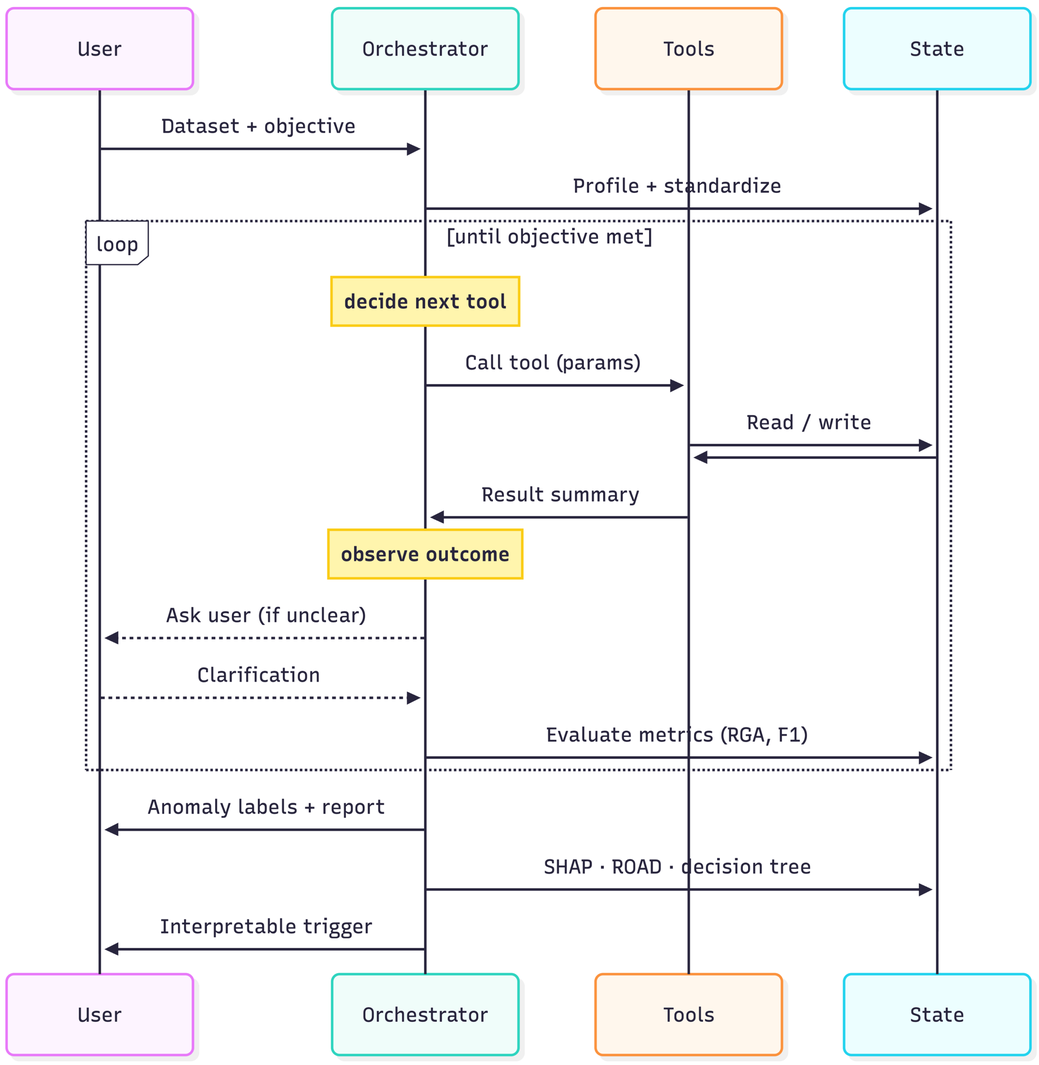
- Detecting anomalies reliably is the first step toward fixing them: yet today's detection pipelines are themselves the bottleneck;
- Preprocessing, model choice, and thresholds are all **hand-tuned** for one specific dataset, and the whole pipeline has to be redone when the network, vendor, or data changes [3].

Questions

- Can a reasoning agent autonomously **build**, **tune**, and **explain** an anomaly detection pipeline?
- Choose preprocessing, model, and thresholds with no operator?
- Adapt mid-execution from intermediate metrics?
- Leave a reasoning trail alongside every detection?

Methods

Reasoning agent: An LLM agent [4] orchestrates calls to tools from a registered toolbox, in an observe → decide → act loop [5]

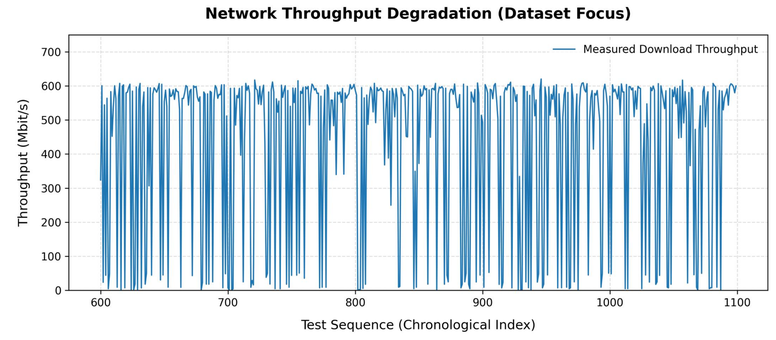


State management: semantic, working and episodic memory

Deterministic / agentic separation:
The LLM reasons and decides; it never computes.
All quantifiable operations (VIF, SHAP, model training, residual analysis) live in deterministic tools, while the agent only orchestrates them. This keeps the system cheap, reproducible, and auditable.

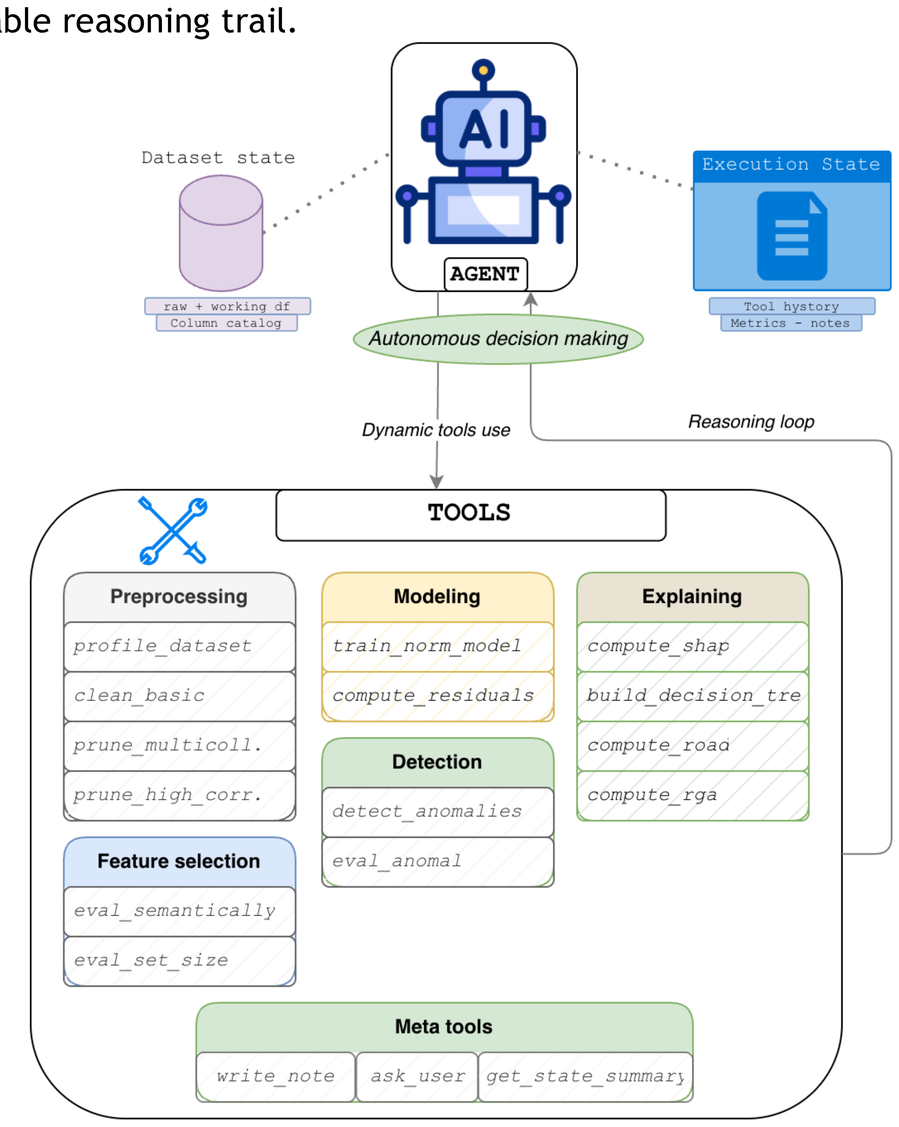
Use case and dataset

We collect network data by setting up an NTD7 server in the lab (https://github.com/m-lab/ndt-server) and running speed tests from a local client configured to impose limitations to simulate network performance anomalies (target: throughput in kbps).



Preliminary Results

BUILD —> agent composes the pipeline from a registered toolbox; new tools plug in without modification.
TUNE —> strategy is revised in flight: residuals and metrics feed back into the next tool call.
EXPLAIN —> every decision logged with tool, inputs, hypothesis → auditable reasoning trail.



Conclusions and future work

- Anomaly detection moves from hand-engineered pipelines to **adaptive, auditable orchestration;**
- The same architecture is the natural substrate for **causal reasoning;**
- Beyond detection, causal network adaptation demands a general, multi-agent architecture and this is its first trigger module; diagnostic agents, safety validation, and closed-loop action come next.

Reference

[1] R. Boutaba et al. *A comprehensive survey on machine learning for networking: evolution, applications and research opportunities.* JISA, 2018.
[2] A. Dethise et al. *Cracking Open the Black Box.* ACM SIGCOMM NetAI Workshop, 2019.
[3] *Machine Learning-Driven Anomaly Detection for 5G O-RAN Performance Metrics.* arXiv:2509.03290, 2025.
[4] Wang et al. A Survey on Large Language Model based Autonomous Agents. Frontiers of Computer Science, 2024.
[5] S. Yao et al. ReAct: Synergizing Reasoning and Acting in Language Models. ICLR, 2023.
[6] Ramirez, Juan et. al.: Interpretable Outlier and Anomaly Detection for Mobile Networks from Small Tabular Data.

Acknowledgments

This work is funded by the European Union Horizon MSCA DN programme FINALITY (G.A. 101168816)

Funded by the European Union

FINALITY

© Institute IMDEA Networks