

Investigating Neuro-Symbolic AI for Context-Aware Process Classification with MITRE ATT&CK® Techniques

Omar Anser, Léo Lavaur, Jérôme Francois

SEDAN team

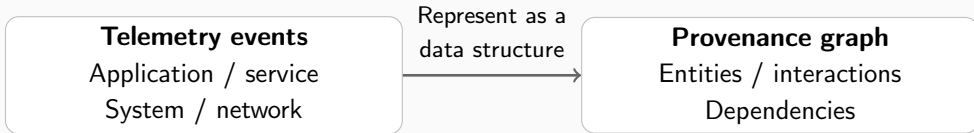
SnT - University of Luxembourg



Luxembourg
National
Research Fund

Causality-Oriented Cybersecurity TELemetry

- Learn causal dependencies between events in microservice infrastructures.
- Use telemetry from application, service, system, and network levels.
- Detect anomalies, identify their causes, and anticipate attacks in microservice infrastructures.



Provenance graphs represent observed activity and its dependencies.

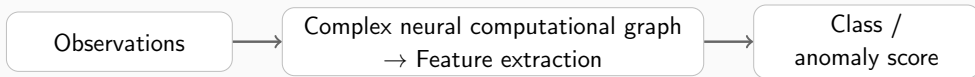
Deep Learning for Provenance-Based Intrusion Detection

- **Masked graph representation learning:** MAGIC [1]

Mainly the graph's structure and the types of entities and interactions.

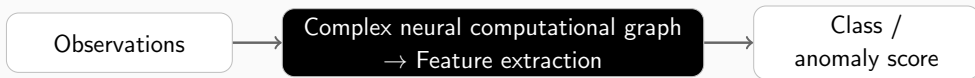
- **Temporal graph neural networks:** KAIROS [2]

Both graph structure and event history (temporal aspect).



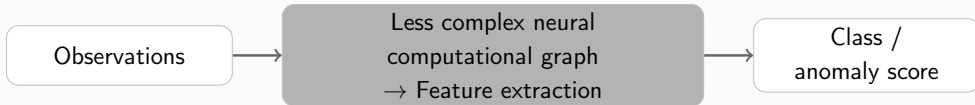
The Complex Computational Graph is a Black Box

- Known complex computational graph $\neq$ Interpretable learned features
- Interpretability: understanding learned features and reasoning; explainability: explaining predictions [3].
- Complex computational graph can mean more tuning [4] + higher compute/memory costs [5].



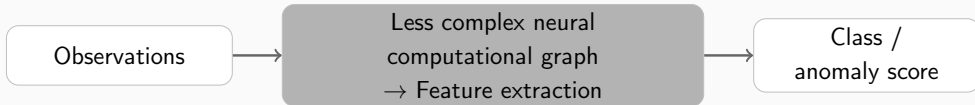
From Black Box to Gray Box?

- How can we integrate domain knowledge into the computational graph so an analyst can understand and modify its contribution to predictions?
- Can this integration also reduce the complexity of the neural computational graph?



From Black Box to Gray Box?

- How can we integrate domain knowledge into the computational graph so an analyst can understand and modify its contribution to predictions?
- Can this integration also reduce the complexity of the neural computational graph?

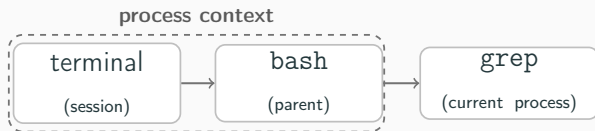


Our study at FMANO — SIGCOMM 2026

**Investigating Neuro-Symbolic AI for Context-Aware Process
Classification with MITRE ATT&CK® Techniques**

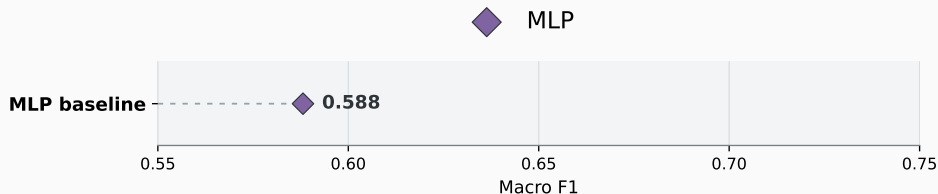
Process Context and ATT&CK® Classification

- Classify attack-related processes into ATT&CK® techniques.
- The same command can map to different techniques depending on its execution context.
- Domain knowledge is encoded as explicit rules linking context to technique labels.



Starting Simple: MLP Without Rules

- CasinoLimit [1]: attack-related processes dataset, labeled with ATT&CK® techniques.
- Convert each process into tokens.
- Predict the ATT&CK® technique from **local evidence only**.



[1] S. Kilian et al., *CasinoLimit: An Offensive Dataset Labeled with MITRE ATT&CK Techniques*, RAID 2025.

Ambiguous Cases: Local Process Cues Are Not Enough

- **LinPEAS** is a Linux privilege-escalation scanner; it runs `grep` to search system files for weaknesses.

LinPEAS

1,094 processes

Observed process: `grep`

Models
predict

System Information
Discovery

Expected

Exploitation for
Privilege Escalation

- **Ligolo-ng** is a tunneling tool; `ping` is used during its session to test connectivity through the tunnel.

Ligolo-ng

40 processes

Observed process: `ping`

Models
predict

Network Service
Discovery

Expected

Protocol Tunneling

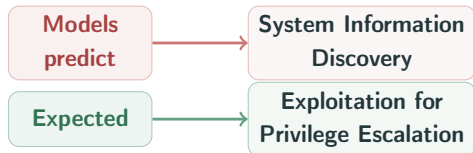
Ambiguous Cases: Local Process Cues Are Not Enough

- **LinPEAS** is a Linux privilege-escalation scanner; it runs `grep` to search system files for weaknesses.

LinPEAS

1,094 processes

Observed process: `grep`

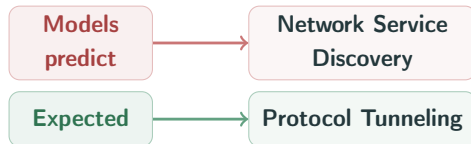


- **Ligolo-ng** is a tunneling tool; `ping` is used during its session to test connectivity through the tunnel.

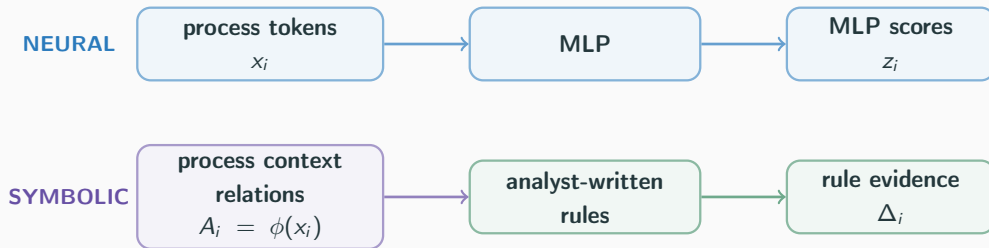
Ligolo-ng

40 processes

Observed process: `ping`



Adding Explicit Contextual Knowledge



Ligolo-ng rule

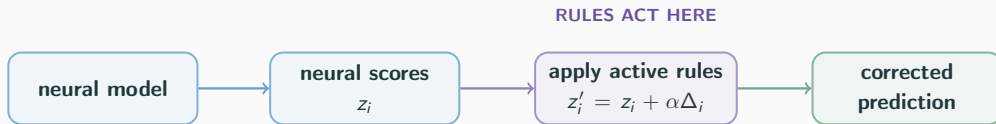
ping linked to an active Ligolo-ng session
+ **Protocol Tunneling**
– Network Service Discovery

LinPEAS rule

grep linked to an active LinPEAS run
+ **Exploitation for Privilege Escalation**
– System Information Discovery

LR-INF: Correct at Prediction Time

Idea: apply expert rules after the neural model, only for the current prediction.



Model change

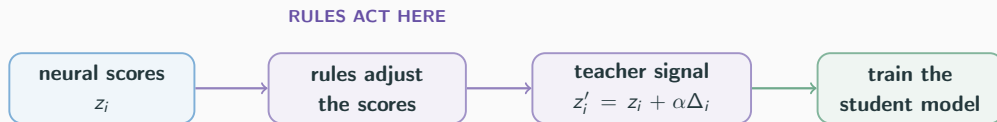
None. The neural parameters remain fixed.

Deployment

Neural model + rule engine.

LR-KD: Learn from Rule-Corrected Targets

Idea: use rules during adaptation to create a teacher for a neural student.



Model change

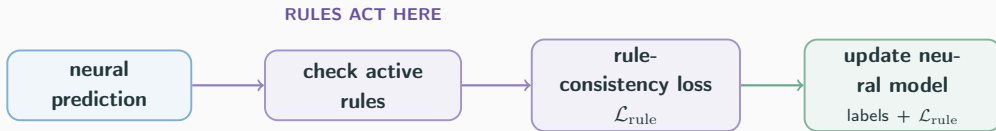
The student learns from labels and rule-corrected model scores.

Deployment

Student model only; rules are no longer required.

LR-E2E: Learn with a Rule-Consistency Loss

Idea: penalize predictions that conflict with active rules and update the model directly.



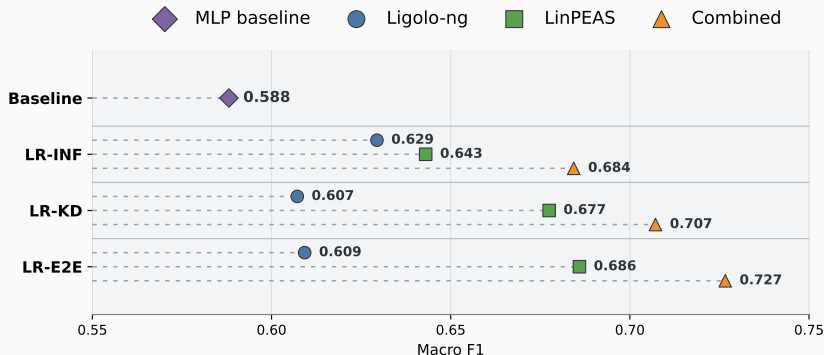
Model change

Neural parameters are updated by label and rule losses.

Deployment

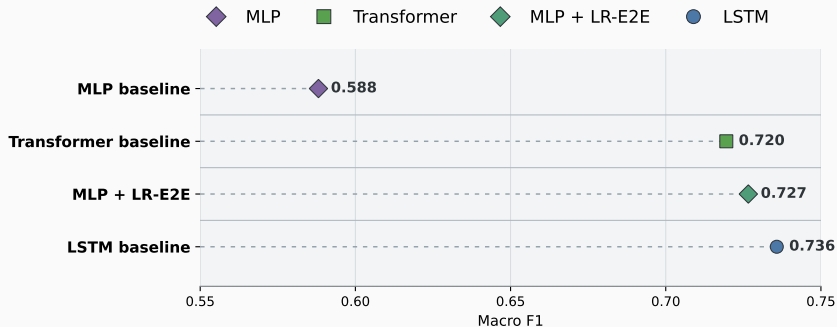
Adapted neural model only; rules are no longer required.

MLP + Rules: Main Result



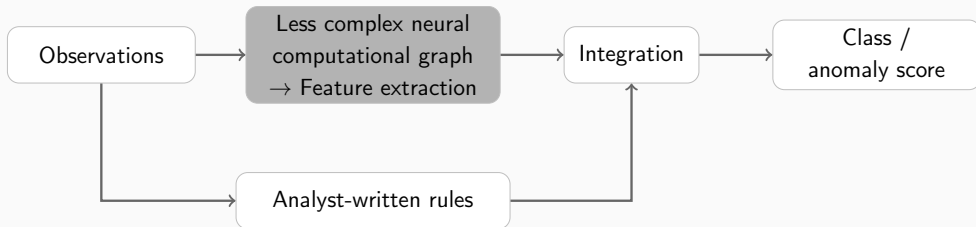
- All rule integrations improve over the **0.588** MLP baseline; combined rules give the largest gains.
- Training-time integration goes beyond the two LR-INF-fixed cases: representation learning learns the rule signal.

Context-Native Models as Comparison



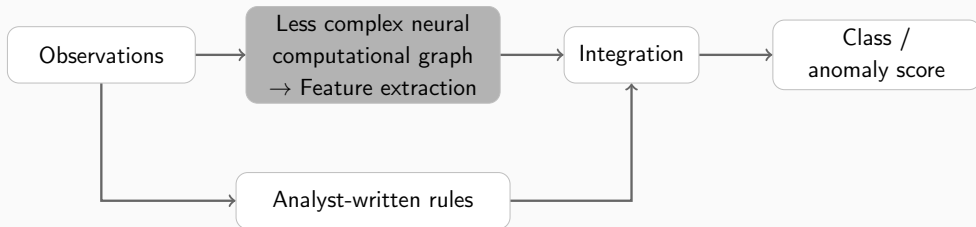
- MLP + LR-E2E scores higher than the Transformer and slightly lower than the LSTM.
- LSTM and Transformer miss the ambiguous cases.

Proof of Concept and Current Limitations



A proof of concept: integrating neural computation and explicit domain knowledge

Proof of Concept and Current Limitations



A proof of concept: integrating neural computation and explicit domain knowledge

But:

- The current work covers only two contextual cases.
- The rules are written manually.
- We lack data to evaluate additional contextual cases and learn the rules automatically.

- Extend the proof of concept using available provenance datasets.
- Study more cases where predictions depend on graph context.
- Learn rules that capture domain knowledge from data.
- Aim for interpretable decisions with lower computational cost.

Apply the Same Principle to Provenance

