TU Delft

The Best of Both Worlds: Fixing Data-Driven NIDS with Neuro-Symbolic Logic

Dr. Andrea Agiollo

PostDoctoral Researcher @ TU Delft

March 19, 2026



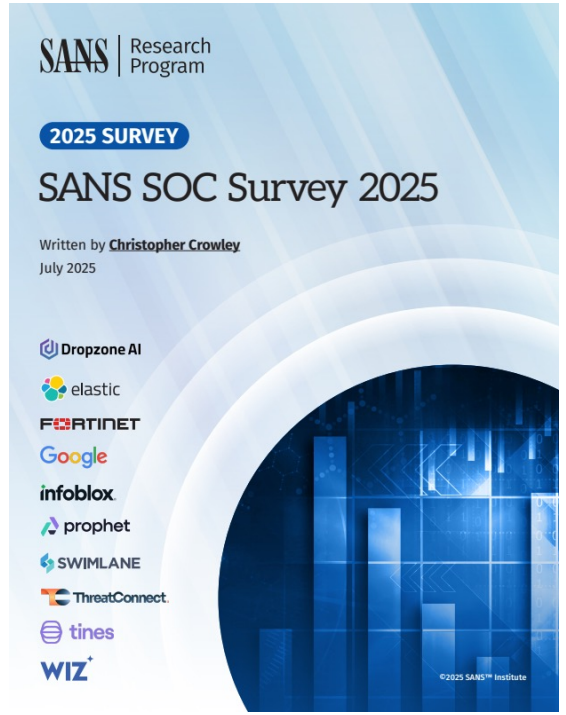
Who am I?

1. MSc in Telecommunications Engineering @ University of Padova, Italy
2. PhD @ University of Bologna, Italy
   - Resource-constrained AI solutions
   - Neuro-Symbolic approaches for resource minimization
   - AI for cybersecurity
3. Visiting PhD @ TU Delft
4. Visiting PhD @ Purdue University
5. PostDoc @ TU Delft, The Netherlands
   - Neuro-Symbolic approaches for cybersecurity
   - Security of Neuro-Symbolic integration
   - Security of Machine Learning



Are ML-NIDS used in practice?

- ML-based NIDS show high detection accuracy in many scientific papers
- Still most SOC rely on rule-based NIDS and ML is thought as “end of pipe” solution
- According to SANS SOC Survey 2025:
  1. Only 30% formally use AI/ML tools in the SOC
  2. 42% of SOCs using AI/ML tools don’t tune or customize the tools
  3. AI/ML is at the bottom of the tools satisfaction list
- Used mostly for anomaly detection (e.g., establishing a baseline of normal network behavior and flagging deviations) with very simple ML models



Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules

Mathew Vermeer
m.vermeer@tudelft.nl
Delft University of Technology

Natalia Kadenko
n.i.kadenko@tudelft.nl
Delft University of Technology

Michel van Eeten
m.j.g.vaneeten@tudelft.nl
Delft University of Technology

Carlos Gañán
c.hernandezganan@tudelft.nl
Delft University of Technology

Simon Parkin
s.e.parkin@tudelft.nl
Delft University of Technology

A quick comparison

Rule-based IDS

Rely on rule reasoning

- High precision for known threats
- Fast and efficient
- Highly interpretable
- Blind to zero-day attacks
- Rigid and easily evaded
- High maintenance burden

ML-based IDS

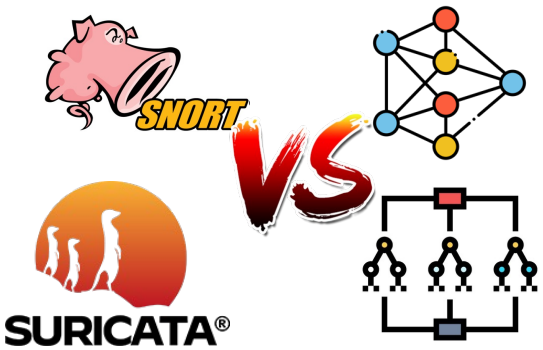
Rely on pattern recognition

- Can detect unknown threats
- Handles complex data
- Lower maintenance burden
- High false positive rate
- The transparency problem
- Resource intensive

	Rule-based IDS	ML-based IDS
Zero-day threat detection	✗	✓
False positives	✓	✗
Interpretability	✓	✗
System maintenance	✗	✓
Computational cost	✗	✓

? Is it possible to achieve the best of both worlds? Any opinion on how to?

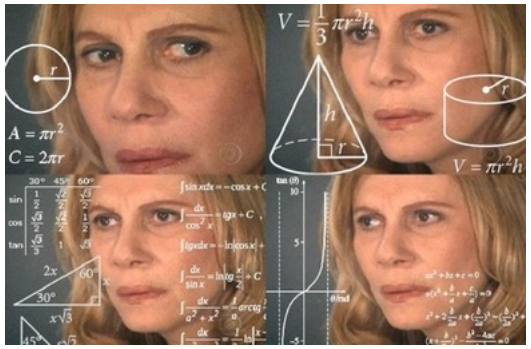
? In which other fields can we see a similar duality? Any guess?

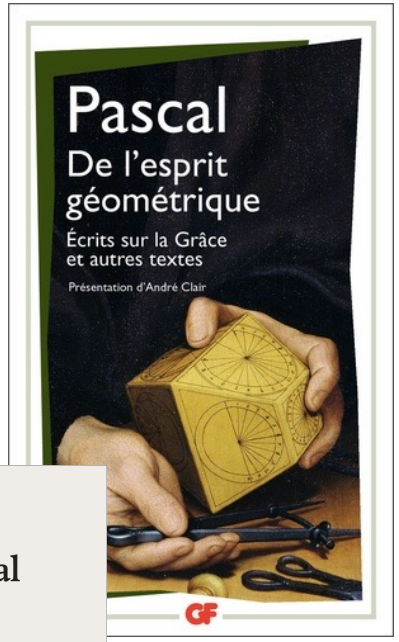


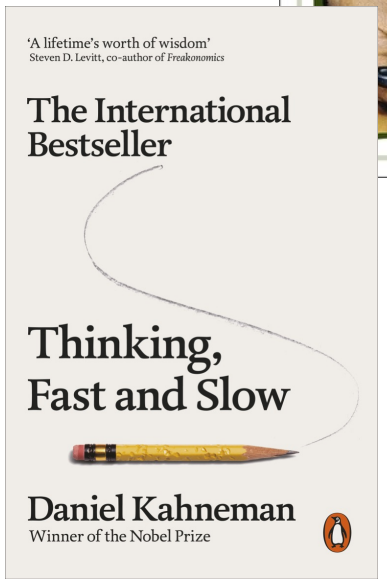
Long standing Dualism in Psychology

- Intuition vs. Reasoning:
  - Esprit de finesse vs. Esprit de géométrie in Philosophy
  - Cognitive vs. Behavioural Psychology
  - Thinking Fast and Slow in Psychology
- Fast intuitive brain:
  - Automatic
  - Pattern-matching
  - Unconscious
- Slow deliberate brain:
  - Analytical
  - Logical
  - Conscious





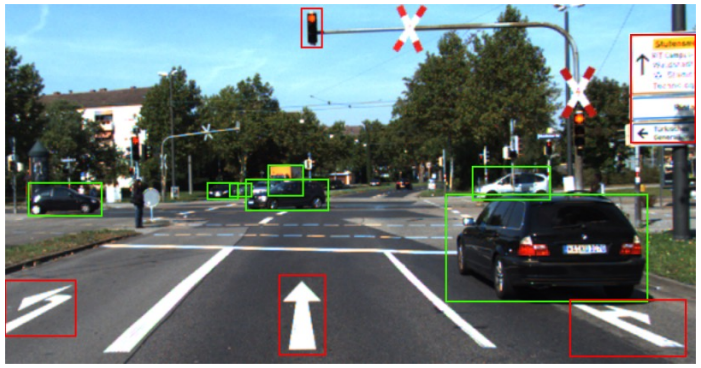




Thinking Fast and Slow

Real-life problems involve two important aspects:

1. Intuitive perception of patterns coming from senses: what do I have in front of me?



Red traffic light

Black car on the right

Three lanes

Pedestrian waiting

Car crossing

2. Reasoning: how should I drive?



Long Standing Dualism in AI

Sub-symbolic AI ("Intuitive" Engine)

- Fast, intuitive, pattern-matching
- Primarily neural networks
- Processes information at a level below explicit human concepts
- Impossible to point to a single node and say *"This is the concept of a dog"*

✓ Advantages:

- Adaptability
- Effective learning from raw data
- Robustness to noisy/unstructured information

✗ Disadvantages:

- Opaque/interpretability
- Lack of guarantees on generalizability
- Data hungriness
- Security issues



Symbolic AI ("Rational" Engine)

- Slow, rational and verifiable
- Primarily logic programming, ontologies, expert systems, etc.
- Processes information using discrete, explicit symbols and rules
- Knowledge is represented by symbols

✓ Advantages:

- Transparent reasoning
- Verifiable
- No training data required

✗ Disadvantages:

- Knowledge acquisition bottleneck
- Missing flexibility
- Not applicable to large unstructured information

Enters Neuro-Symbolic AI

- Symbolic and Sub-Symbolic approaches are complementary
- Advantages of one cover disadvantages of the other
- **Rough definition:** *a hybrid AI paradigm that integrates symbolic reasoning (logic, rules, knowledge graphs) with neural learning (pattern recognition) to enable systems that can both learn from data and reason about it.*
- Neuro-symbolic AI aims:
  - Learn from raw data like images or text (neural networks)
  - Reason about abstract concepts and relationships (symbolic logic)
  - Explain decisions in human-understandable terms

DeepProbLog:
Neural Probabilistic Logic Programming

A Semantic Loss Function for Deep Learning with Symbolic Knowledge

Jingyi Xu [1] Zilu Zhang [2] Tal Friedman [1] Yitao Liang [1] Guy Van den Broeck [1]

DeepStochLog: Neural Stochastic Logic Programming

Thomas Winters,*[1] Giuseppe Marra,*[1] Robin Manhaeve,[1] Luc De Raedt[1,2]

[1]Department of Computer Science; Leuven.AI, KU Leuven, Belgium,

Logic Tensor Networks

Samy Badreddine [a,b,*], Artur d'Avila Garcez [c], Luciano Serafini [d], Michael Spranger [a,b]

[a] *Sony Computer Science Laboratories Inc, 3-14-13 Higashigotanda, 141-0022, Tokyo, Japan*
[b] *Sony AI Inc, 1-7-1 Konan, 108-0075, Tokyo, Japan*
[c] *City, University of London, Northampton Square, EC1V 0HB, London, United Kingdom*
[d] *Fondazione Bruno Kessler, Via Sommarive 18, 38123, Trento, Italy*

DeepProofLog: Efficient Proving in Deep Stochastic Logic Programs

Ying Jiao*[1], Rodrigo Castellano Ontiveros*[2], Luc De Raedt[1,3], Marco Gori[2], Francesco Giannini[4,5], Michelangelo Diligenti[2], Giuseppe Marra[1]

[1]KU Leuven, Belgium

HyperTree Proof Search for Neural Theorem Proving

Guillaume Lample*† Marie-Anne Lachaux*† Thibaut Lavril*† Xavier Martinet*†

Amaury Hayat§ Gabriel Ebner† Aurélien Rodriguez† Timothée Lacroix*†

Towards Neural Theorem Proving at Scale

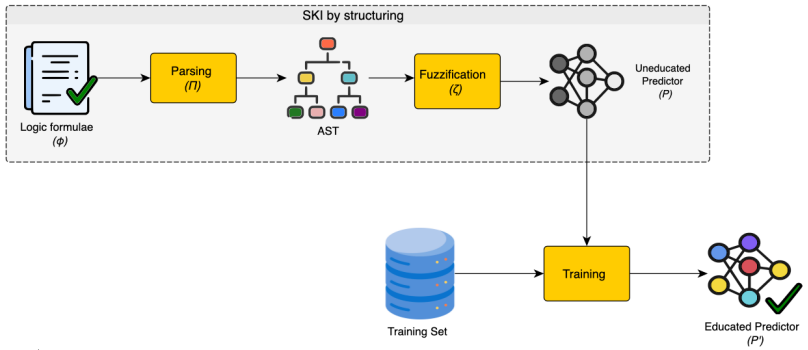
aly

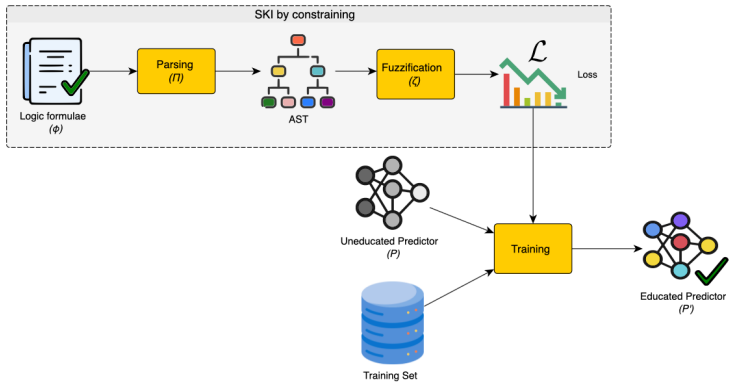
si.it, francesco.giannini@sns.it

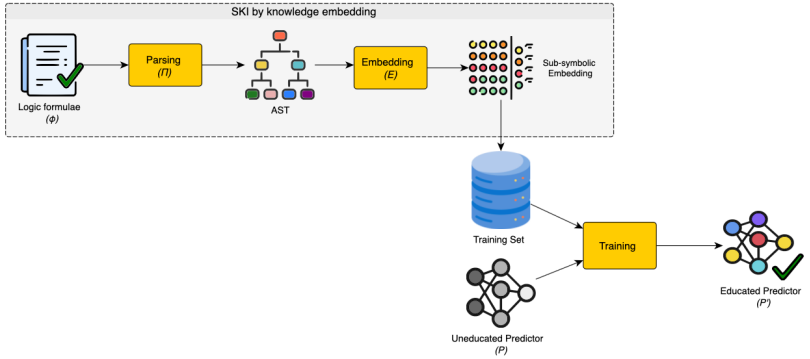
Pasquale Minervini*[1] Matko Bosnjak*[1] Tim Rocktäschel[2] Sebastian Riedel[1,3]

How does NeSy work?

- We proposed a taxonomy focusing specifically on how logic is integrated in NeSy systems:
  - Predictor structuring: (a part of) a sub-symbolic predictor is created to mirror the symbolic knowledge via its own internal structure. NNs whose internal structure is crafted to represent logic predicates via neurons, and logic connectives via synapses.
  - Guided learning: symbolic knowledge is used to steer the learning process of ML predictors by either penalizing inconsistent behaviors or by incentivizing consistent behaviors about the symbolic knowledge.
  - Knowledge embedding: symbolic knowledge converted into numeric-array form to be provided as 'ordinary' input for the sub-symbolic predictor undergoing injection.

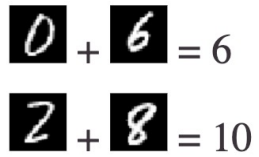






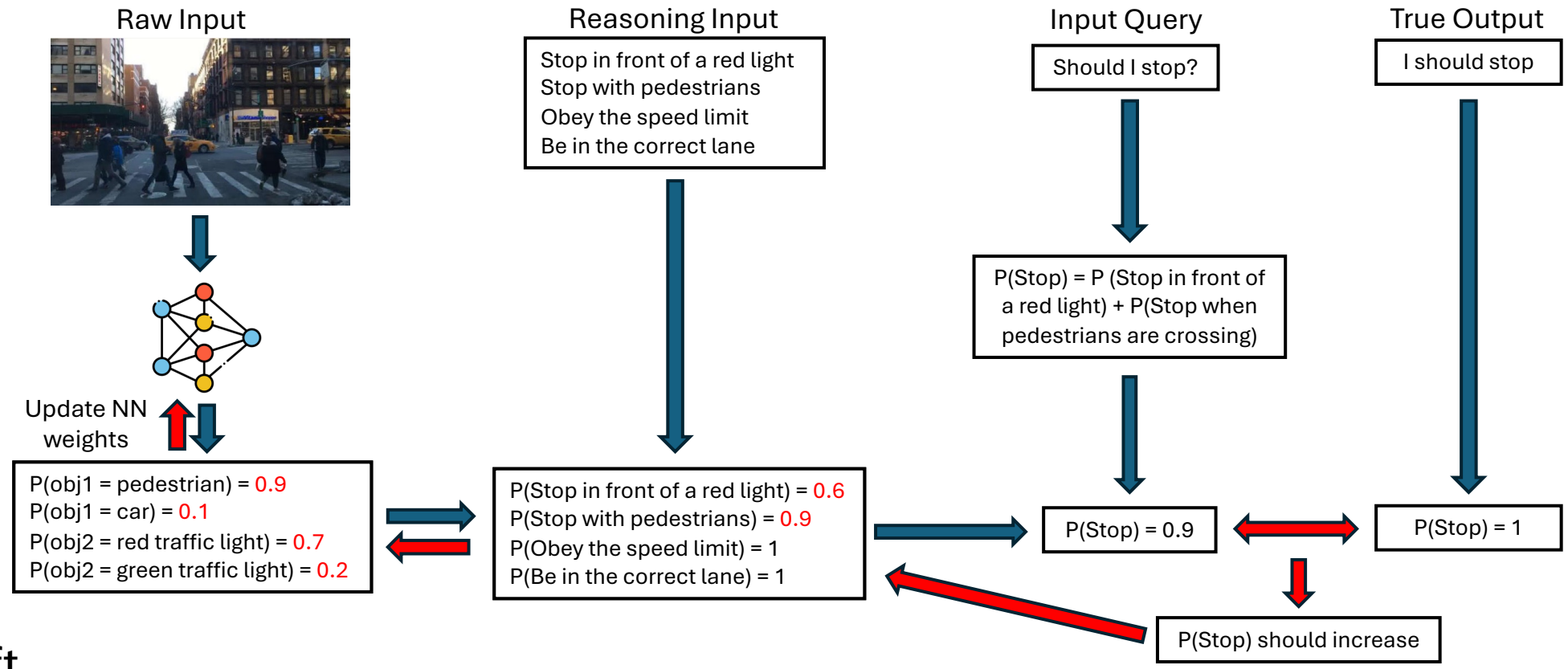
Ciatto, G., Sabbatini, F., **Agiollo, A.**, Magnini, M., & Omicini, A. (2024). "Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review". ACM Computing Surveys, 56(6), 1-35.

NeSy Example: DeepProbLog



- Toy example: handwritten digits addition
- NN recognize the digits and write logic to compute the sum
- DeepProbLog version embeds NN into logic program as neural predicates

```
nn(m_digit, [X], Y, [0...9]) :: digit(X,Y).
addition(X,Y,Z) :- digit(X,N1), digit(Y,N2), Z is N1+N2.
```



Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., & De Raedt, L. (2018). “Deepproblog: Neural probabilistic logic programming”. Advances in neural information processing systems, 31.

Why should we care about NeSy?

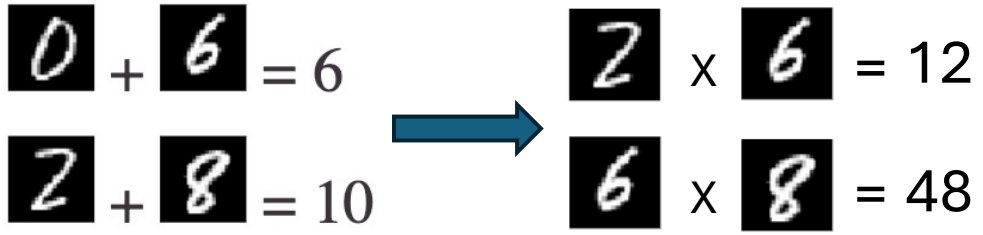
General case:

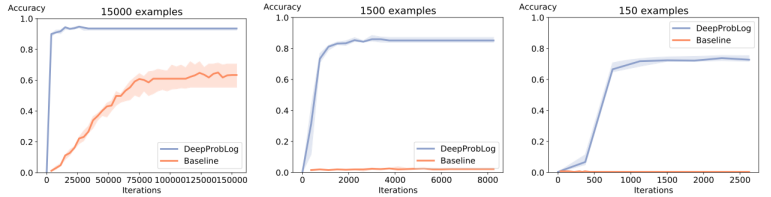
1. End-to-end learning and reasoning
2. More flexible
3. More data-efficient
4. More robust
5. More transparent

For the NIDS context:

1. Some traffic patterns are difficult to encode with pure logic
2. Part of traffic analysis must be deterministic (e.g., attack targets always specific port, contains specific string, etc.)

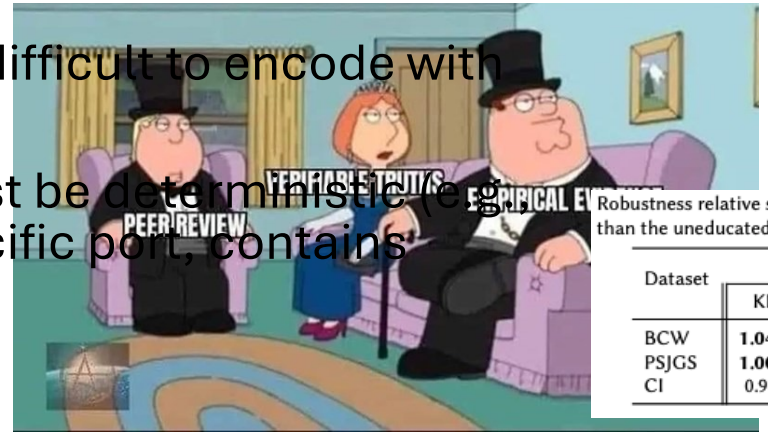






(a) 15 000 examples (b) 1 500 examples (c) 150 examples

Fig. 7. MNIST Multi-Digit Addition (**T2**). The graphs show the accuracy on the validation set during training for different training set sizes.



Dataset	Model	Set	Data efficiency (KB)	Energy (mWh)	Memory (FLOPs)	Latency (ms)	Accuracy (%)
CI	Uneducated		–	–	–	–	84.63
	KBANN	Train	1653.79	−1.41	– 2468	−0.43	84.78
		Test		−0.02			
	KILL	Train	4016.90	−0.70	4200	0	84.81
		Test		0			
	KINS	Train	4263.50	−1.41	-6220	−0.44	84.77
		Test		−0.02			

Robustness relative scores. Bold numbers are the ones greater than 1 (i.e., the educated model is more robust than the uneducated one).

Dataset	$R_{N,D}(\mathcal{I})$ drop			$R_{N,D}(\mathcal{I})$ noise			$R_{N,D}(\mathcal{I})$ flip		
	KINS	KILL	KBANN	KINS	KILL	KBANN	KINS	KILL	KBANN
BCW	**1.0493**	**1.0318**	**1.0382**	0.9960	0.9985	**1.0109**	0.9994	**1.0184**	0.9520
PSJGS	**1.0045**	0.9968	0.8425	0.9950	0.9984	**1.0145**	0.9962	**1.0026**	**1.6749**
CI	0.9998	**1.0039**	**1.0043**	0.9992	**1.0012**	0.9965	0.9897	**1.1703**	0.9815

Agiollo, A., Rafanelli, A., Magnini, M., Ciatto, G., & Omicini, A. (2023). “Symbolic knowledge injection meets intelligent agents: QoS metrics and experiments”. Autonomous Agents and Multi-Agent Systems, 37(2), 27.

Rafanelli, A., Magnini, M., **Agiollo, A.**, Ciatto, G., & Omicini, A. (2024). “An empirical study on the robustness of knowledge injection techniques against data degradation”. In CEUR Workshop Proceedings (Vol. 3735, pp. 20-32).

Towards NeSy NIDS



Q1: What can we do with NeSy in NIDS?

- SQ1: What reasoning behavior to be enforced with the logic component?
- SQ2: What pattern matching ability we want to learn directly from the data?
- SQ3: What NeSy system?

Working on 3 perspectives:

1. DeepChronos: Enforce temporal logic into multi-step attack detection
2. DeepWeave: Use logic to bridge packet-level and flow level detection
3. DeepSNORT: General purpose NeSy NIDS

Q2: What can we do with NeSy in SOCs?

- SQ1: For which SOC tasks can NeSy be helpful?
- SQ2: What NeSy system is the most well-suited?

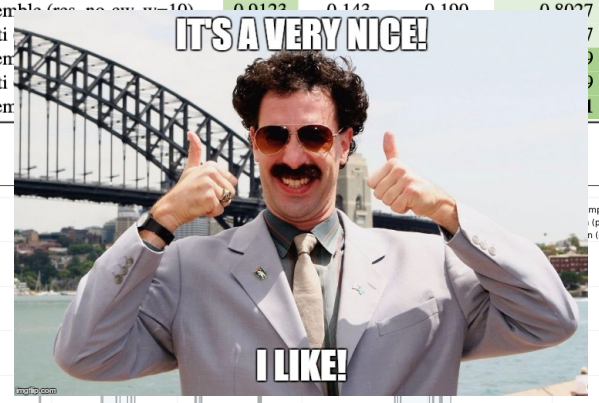
Working on 2 perspectives:

1. Alert fatigue: use NeSy to analyze NIDS alerts and identify false positives grounded on logic
2. VeriSec: A Neuro-Symbolic QA Security System

Introducing DeepChronos



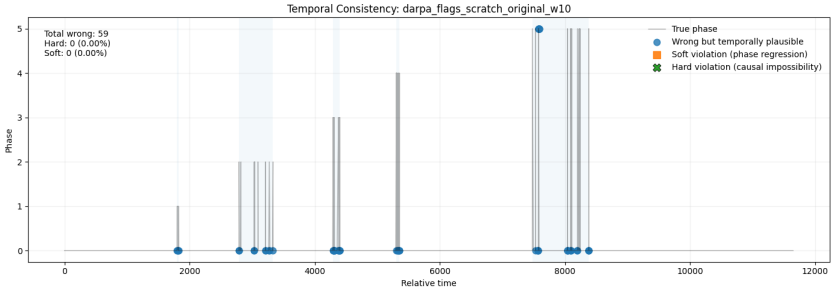
Model	F1	Hard Rate	Soft Rate	Temporal Score
Ensemble (orig, cw, w=10)	**0.9291**	0.000	0.263	0.8764
Ensemble (res, cw, w=10)	0.9129	0.125	0.167	0.8170
Ensemble (res, no cw, w=10)	0.9123	0.143	0.190	0.8027
Multi				
Ensem				
Multi				
Ensem				



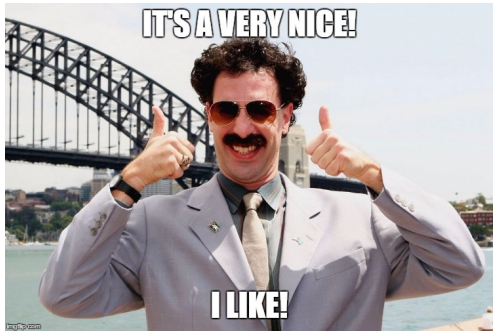
- Focus on multi-step attacks: complex cyberattacks where threat actor executes sequence of distinct, logically connected actions over *a period of time* to achieve objective
- DARPA2000 example: *Reconnaissance* (IP sweep) -> *Probing* (identify vulnerable services) -> *Compromise* (gain root access) -> *Preparation* (deploy DDoS infrastructure) -> *Execution* (launch DDoS)
- ML-based NIDS using temporal NNs fail to preserve temporal consistency
- The NeSy bridge:
  - NNs in charge of detecting each step of a multi-step attack
  - Events are mapped to probabilistic logic facts annotated with timestamps
  - Reasoning over temporal rules such as **Attack(target) :- PortScan(target) @ t1 AND ExploitAttempt(target) @ t2 AND t2 > t1**
  - *Preliminary results:* Temporal consistency fully preserved

```
% Neural networks for each phase
nn(net1, [X], Z, [benign, phase1]) :: phase(1, X, Z).
nn(net2, [X], Z, [benign, phase2]) :: phase(2, X, Z).
nn(net3, [X], Z, [benign, phase3]) :: phase(3, X, Z).
nn(net4, [X], Z, [benign, phase4]) :: phase(4, X, Z).
nn(net5, [X], Z, [benign, phase5]) :: phase(5, X, Z).

multi_step(X, P1, P2, P3, P4, Outcome) :-
    Next is P1 + P2 + P3 + P4 + 1,
    phase(Next, X, Outcome).
```



Introducing DeepWeave



- Focus on detection granularity: most ML-based NIDS either focus on packet-level or flow-level detection
- Packet-level detection catches payload signatures but lacks broader context + computationally heavy (e.g., deep packet inspection)
- Flow-level detection sees overall traffic patterns and volume but is blind to the actual malicious payload signatures
- Both packet-level and flow-level ML-based NIDS do not generalize well outside the lab benchmark/dataset considered for optimization
- The NeSy bridge:
  - Fuse micro-level payload inferences with macro-level flow inferences using logical rules
  - Packet-level NNs analyze raw payload bytes and output packet-level probabilities (0.88 prob of malicious payload)
  - Flow-level NNs analyze network statistics and output flow-level probabilities (0.92 prob of DDoS pattern)
  - Use logic rules to combine neural outputs merging the granularities **High_Risk(flow) :- Flow_Anomalous(flow) AND Contains(flow, packet) AND Packet_Malicious(packet)**
  - *Preliminary results:* NeSy behaves much better than both NNs and static rules at both granularities

Introducing DeepSNORT



- Question: rather than focusing on task-specific solutions, can we make a general NeSy-based IDS similar to SNORT or SURICATA?
- Inspired by DeepProbLog, we are extending SNORT to function as a neural-symbolic reasoner:
    1. Neural (perception): NN(s) process raw packet streams to extract high-level semantic concepts that are hard to manually code in SNORT (e.g., application type, user intent)
    2. Symbolic (reasoning): extracted concepts are fed into SNORT as facts to write *"smart rules"* that combine rigid signatures with flexible, learned patterns
- Intuition: NNs see the traffic and label it ('this looks like a video stream') + SNORT thinks about the policy ('video streams are forbidden on this secure port')
- Ideal rule template: **alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"NeuroSymbolic alert"; flow:established,to_server; NN_1:"semantic_concept_detected"; NN_2:"another_semantic_concept_detected";)**

DeepSNORT tentative examples

Advanced Protocol Tunneling (e.g., SSH over HTTP): Attackers often tunnel forbidden protocols through allowed ones: wrap SSH traffic inside HTTP packets to communicate with a command-and-control server on port 80 or 443.

- SNORT operates heavily on ports and explicit byte sequences: if the outer wrapper of the packet complies with HTTP standards, SNORT likely to see it as legitimate web traffic
- DeepSNORT solution:
  - NN trained on flow to extract the application type concept (SSH) -> even if the traffic is wrapped in HTTP headers flow behavior look like an interactive SSH session
  - Symbolic reasoning: SNORT holds the rigid policy that SSH traffic is strictly forbidden on port 80
- Conceptual DeepSNORT rule: ***alert tcp any any -> any 80 (msg:"Protocol Tunneling - SSH masquerading as HTTP"; neuro_label:"app_type:ssh";)***

DeepSNORT tentative examples

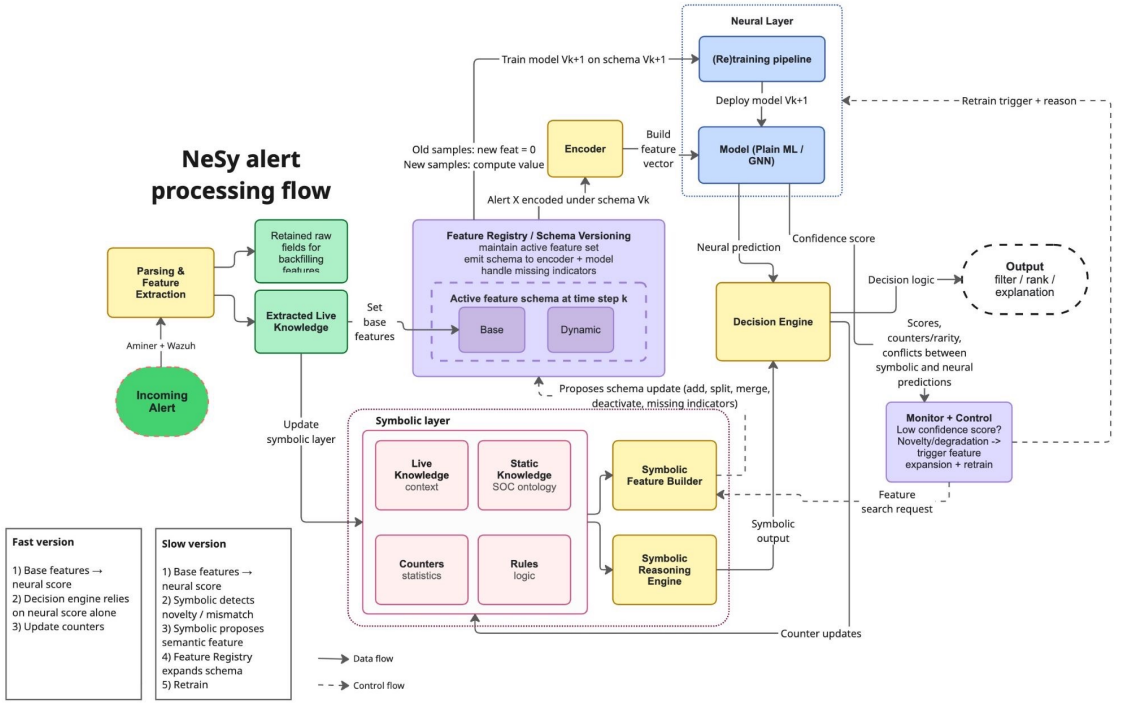
Obfuscated Web Shells and Polymorphic Payloads: attacker uploads a web shell to a compromised server and obfuscate subsequent interactions to encode commands using Base64, XOR encryption, or string concatenation (e.g., passing e x e c separately in a PHP payload) to evade detection.

- SNORT signatures rely on exact string matches (e.g., content:"/bin/sh" or content:"cmd.exe") -> polymorphic or encoded payload is missed by the signature
- DeepSNORT solution:
  - NN is anomaly detector to learn the latent representation of malicious code execution, regardless of encoding used (flags the packet stream with semantic_concept: malicious_command_injection)
  - Symbolic reasoning: SNORT checks web servers and if connection is established
- Conceptual DeepSNORT rule: ***alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Obfuscated Command Injection"; flow:established,to_server; neuro_label:"malicious_command_injection")***

Tackling alert fatigue using NeSy

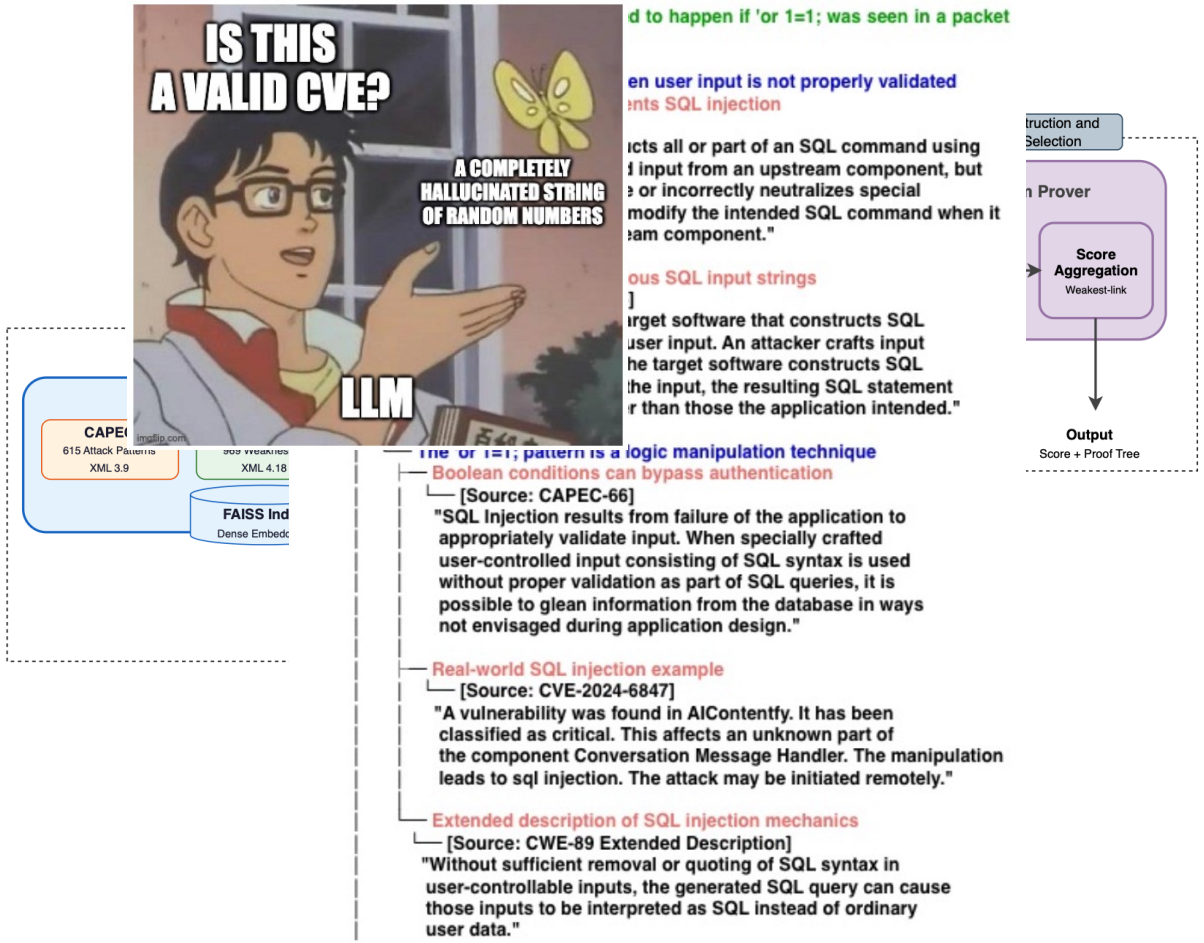


- SOC analysts are overwhelmed by alert fatigue
- Majority of alerts are false positives, leading to missed critical threats and analyst burnout
- Symbolic-based dynamic feature mining continuously adapts to new attack patterns, extracting the most relevant features on the fly
- ML model leverages these dynamic features to accurately classify and filter out false positives
- Preliminary results: mining dynamic features symbolically can reduce false positives under temporal drift



NeSy question answering security system

- LLMs in cybersecurity often suffer from hallucinations, unreliable for question answering
- NeSy system that replaces black-box guessing with human-verifiable reasoning:
  1. Decompose complex security queries into formal, goal-oriented proof trees
  2. Restricts information retrieval exclusively to recognized cybersecurity databases (CVE, CWE, CAPEC)
  3. Explicitly signals to model when knowledge is missing, recognizing its limitations instead of generating plausible but incorrect answers
- 100% grounded transparent audit trail for security analysts

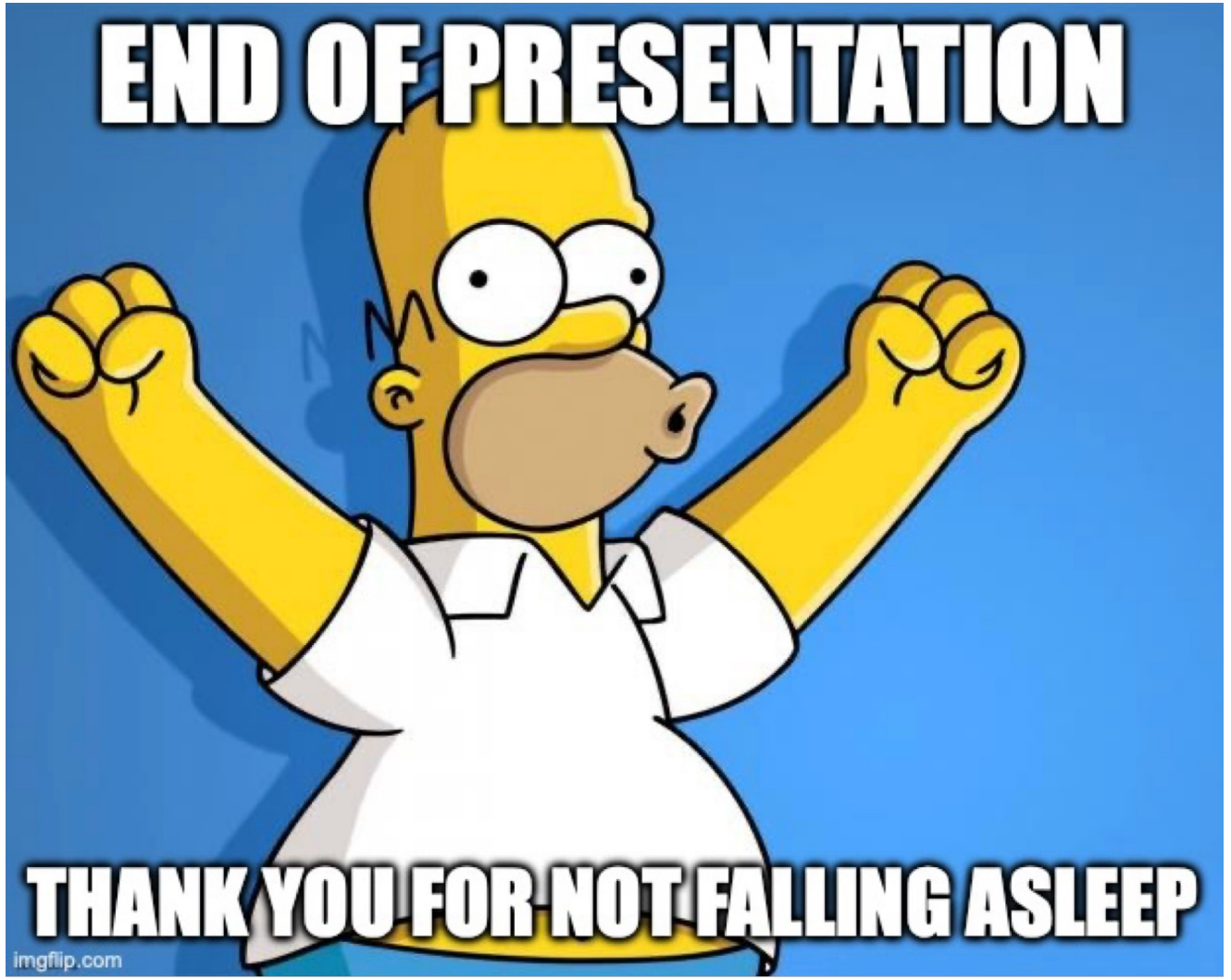


Towards useful NeSy in SOCs

- Any other suggestion on DeepSNORT usage?
- Any suggestions on how to use NeSy in practical SOCs?



Questions?



LinkedIn

GitHub