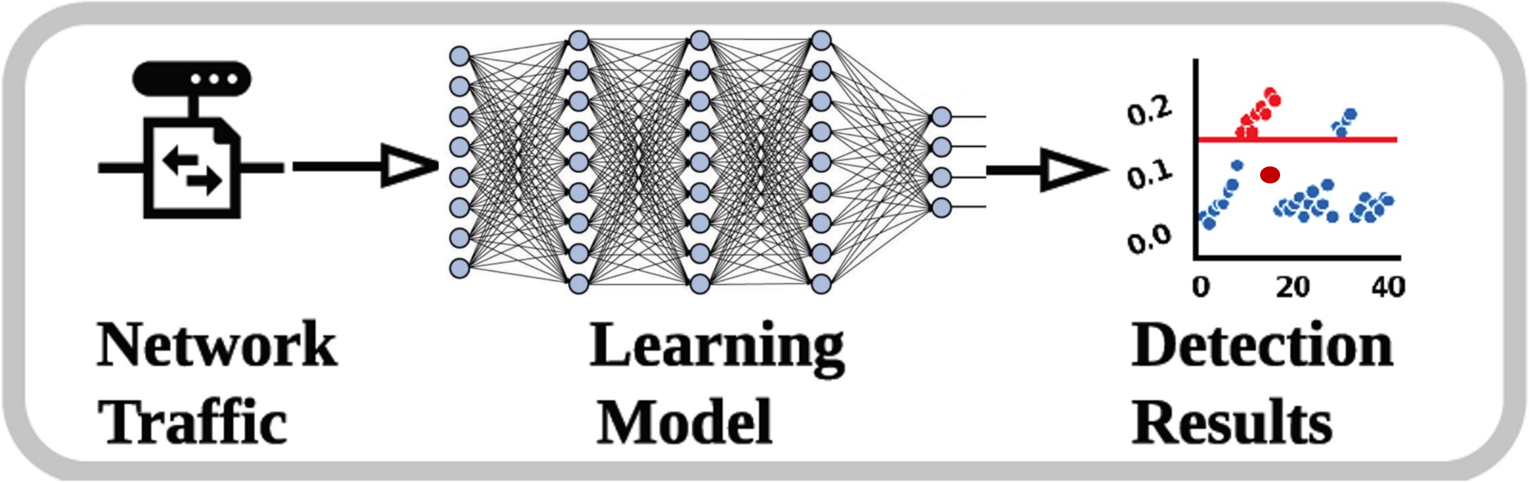
xNIDS: Explaining Deep Learning-based Network Intrusion Detection Systems for Active Intrusion Responses

Feng Wei[1], Hongda Li[2], Ziming Zhao[1] , and Hongxin Hu[1]

1 University at Buffalo The State University of New York

2 paloalto NETWORKS

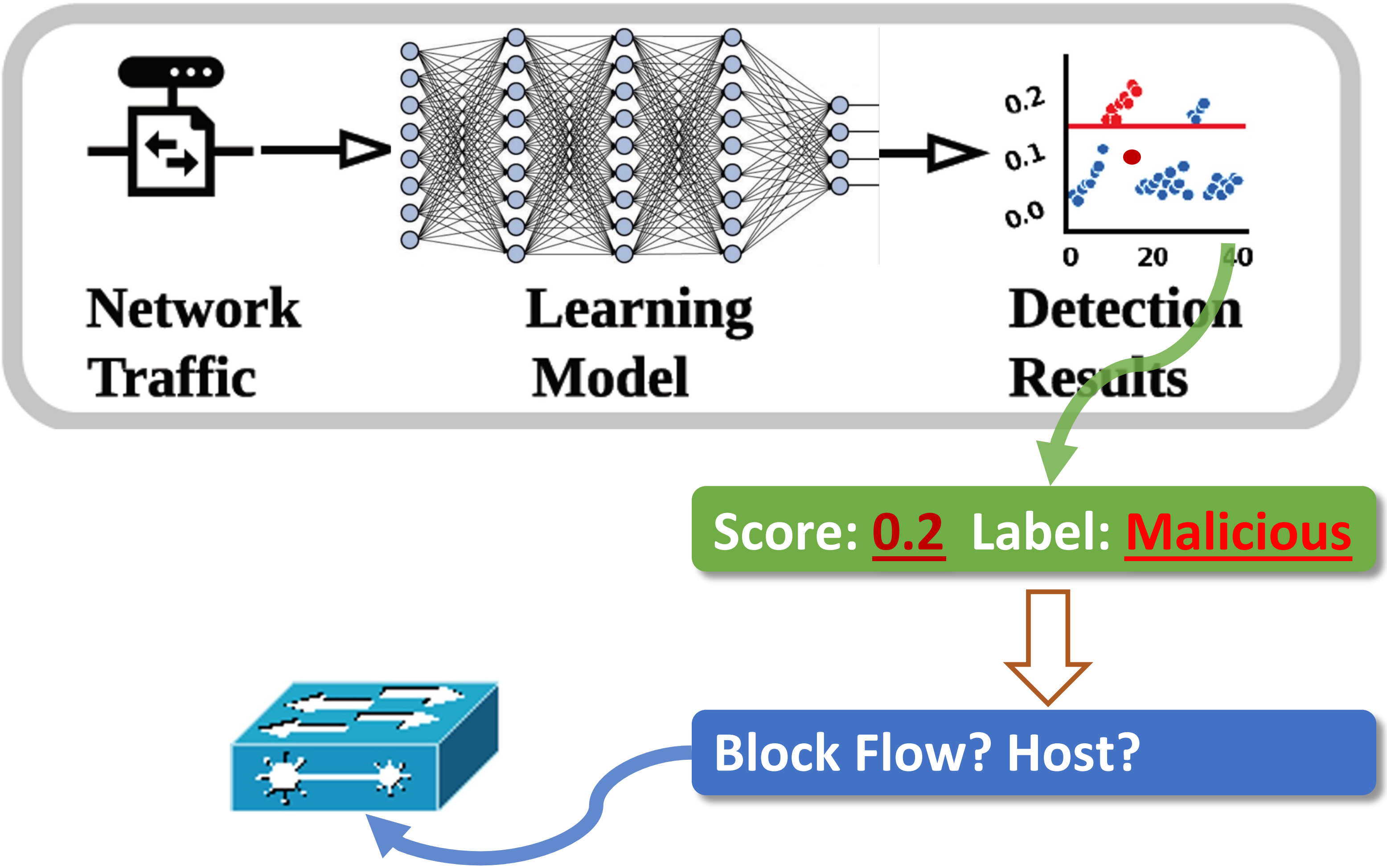
Deep Learning-based Network Intrusion Detection Systems



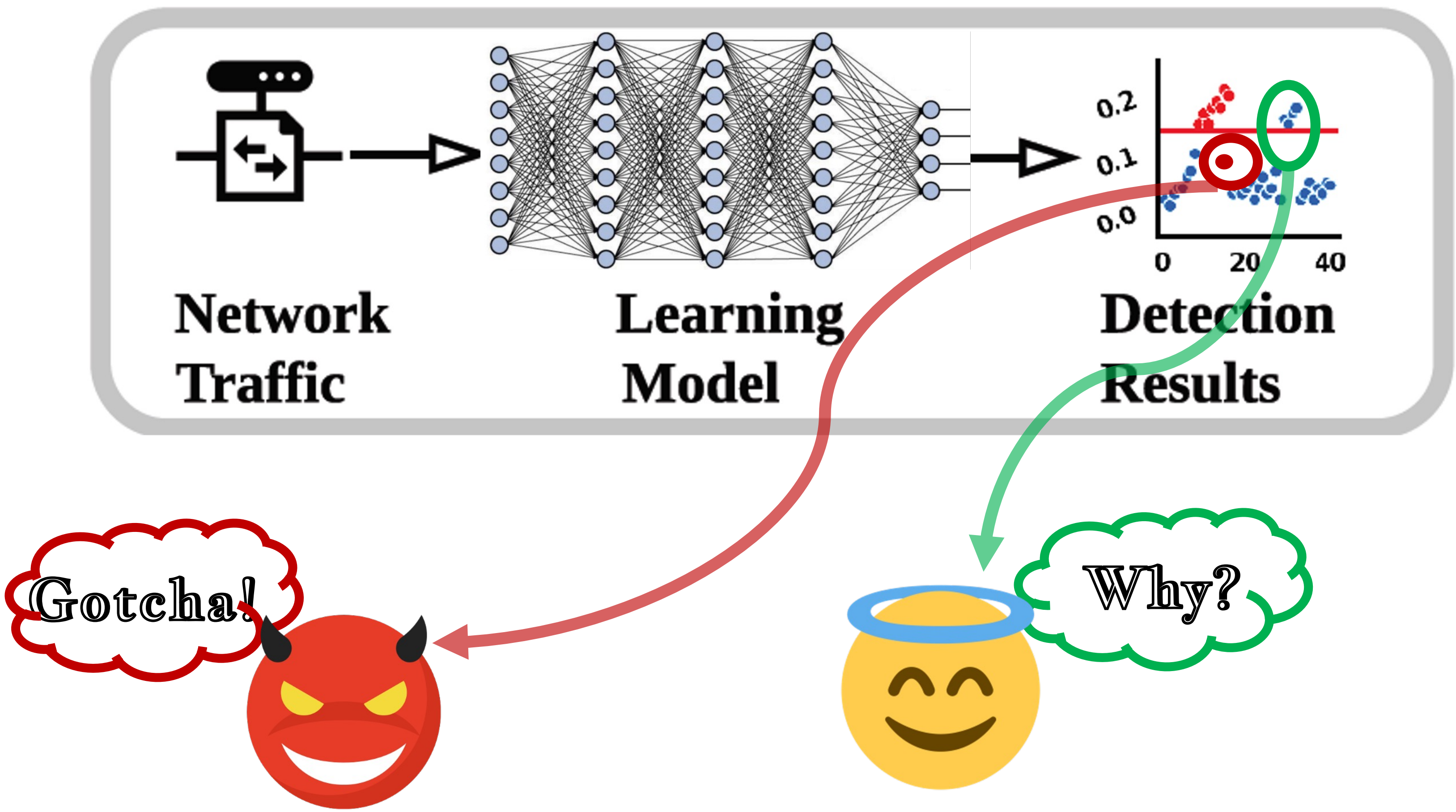
Pros:
- Detect unseen attacks
- Capture complicated patterns

Cons:
- Semantic gap
- High cost of errors

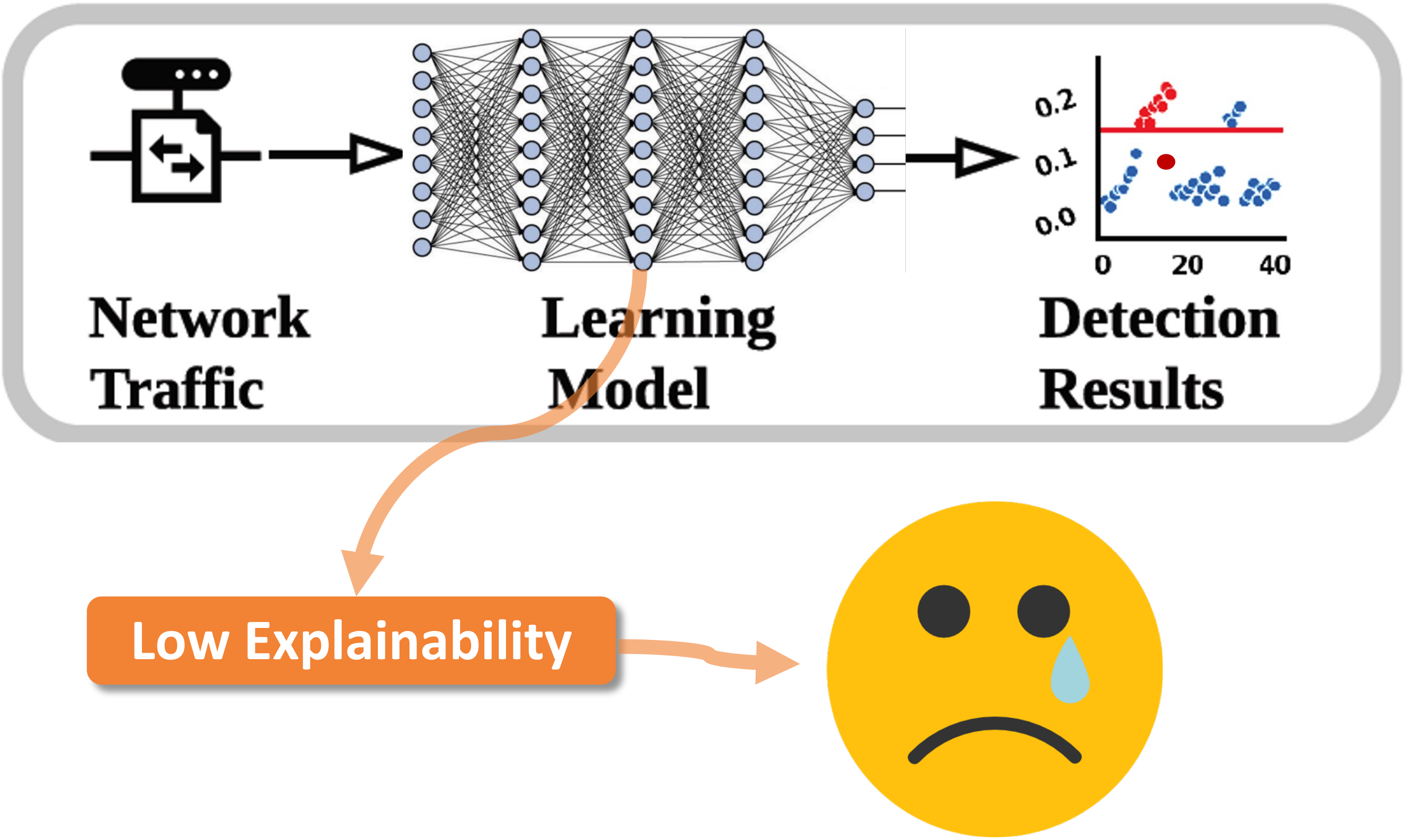
Cons: Semantic Gap



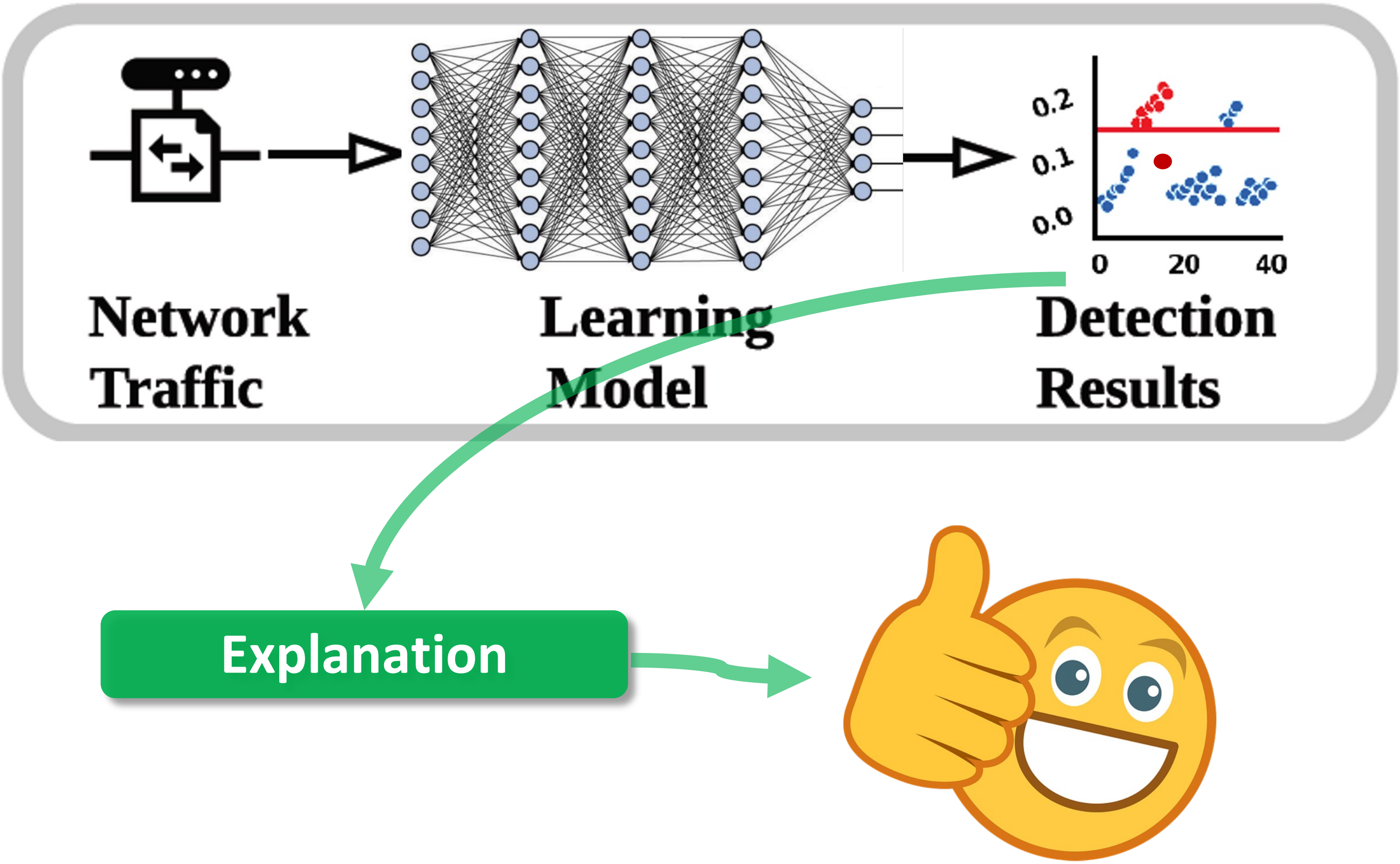
Cons: High Cost of Errors



Root Cause of those Drawbacks

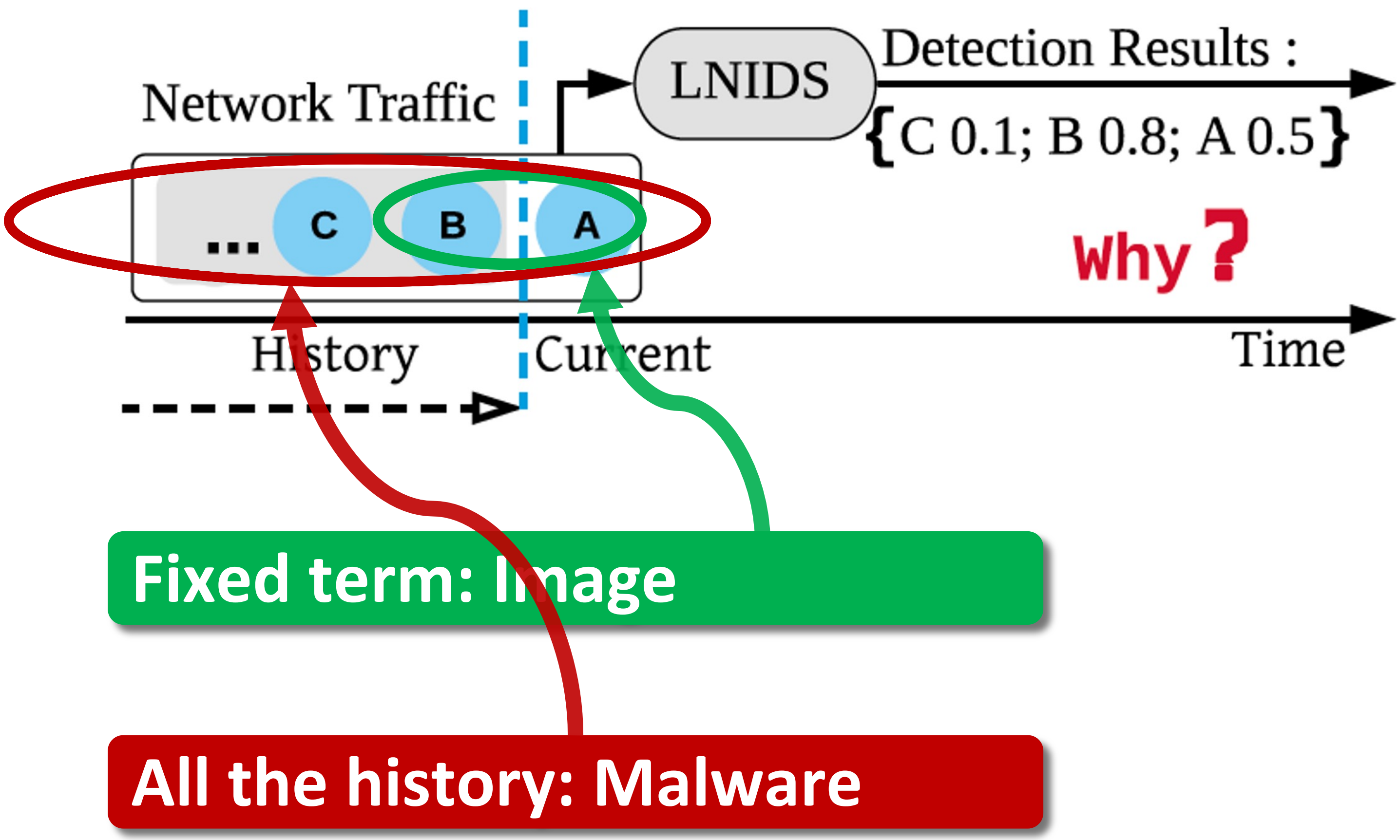


New Trends

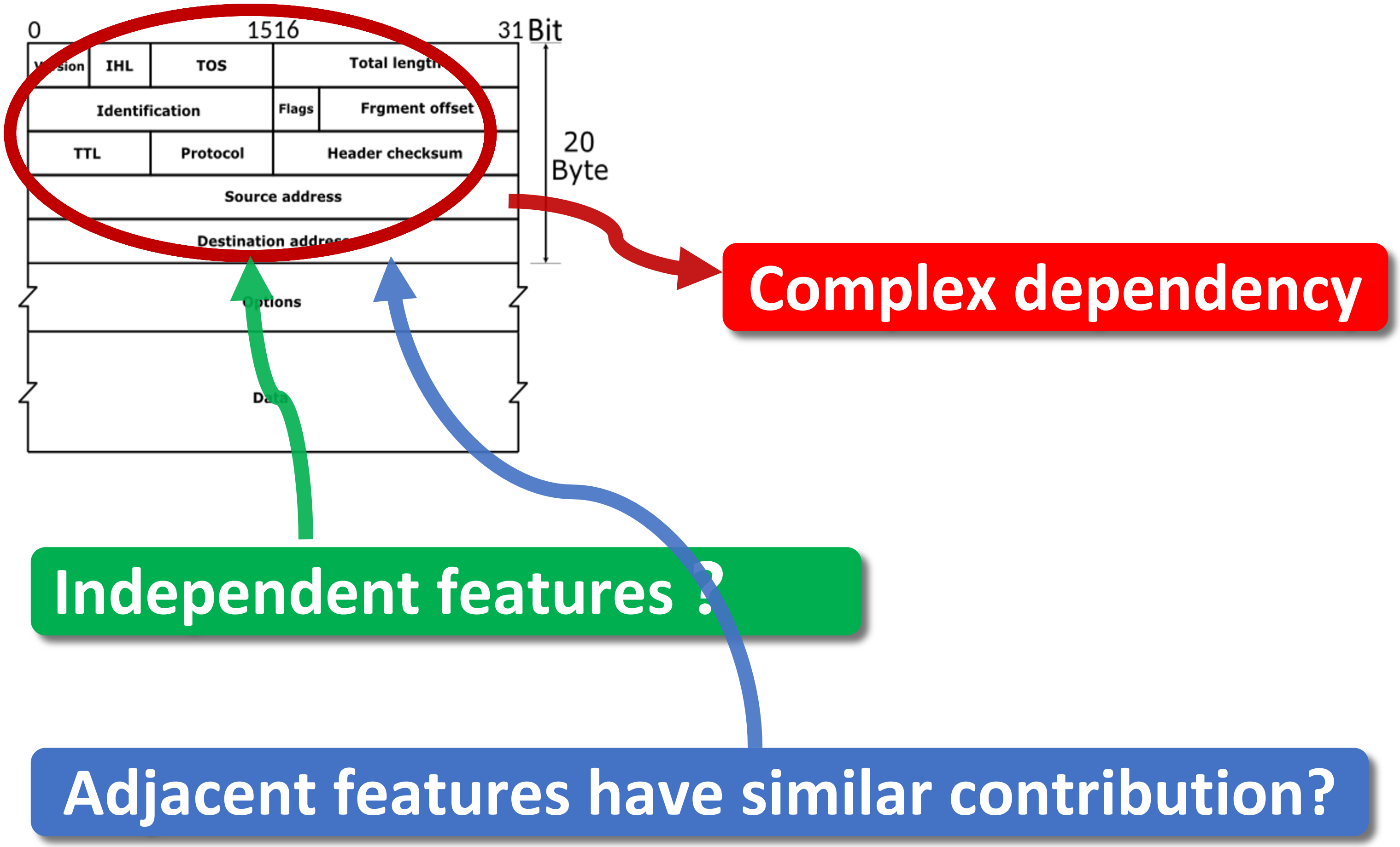


0.2
0.1
0.0
0
20
40
Network Traffic
Learning Model
Detection Results
Explanation

CH1: How to consider history inputs?



CH2: How to capture complex feature dependencies in structured data?



Challenges in Generating Defense Rules

Balance precision and generalization

- **Too specific rules**
  - Overfitting and overwhelming number of rules
- **Too generic rules**
  - Disrupting normal services

Challenges in Generating Defense Rules

Applicable to different defense tools

Similar functionality

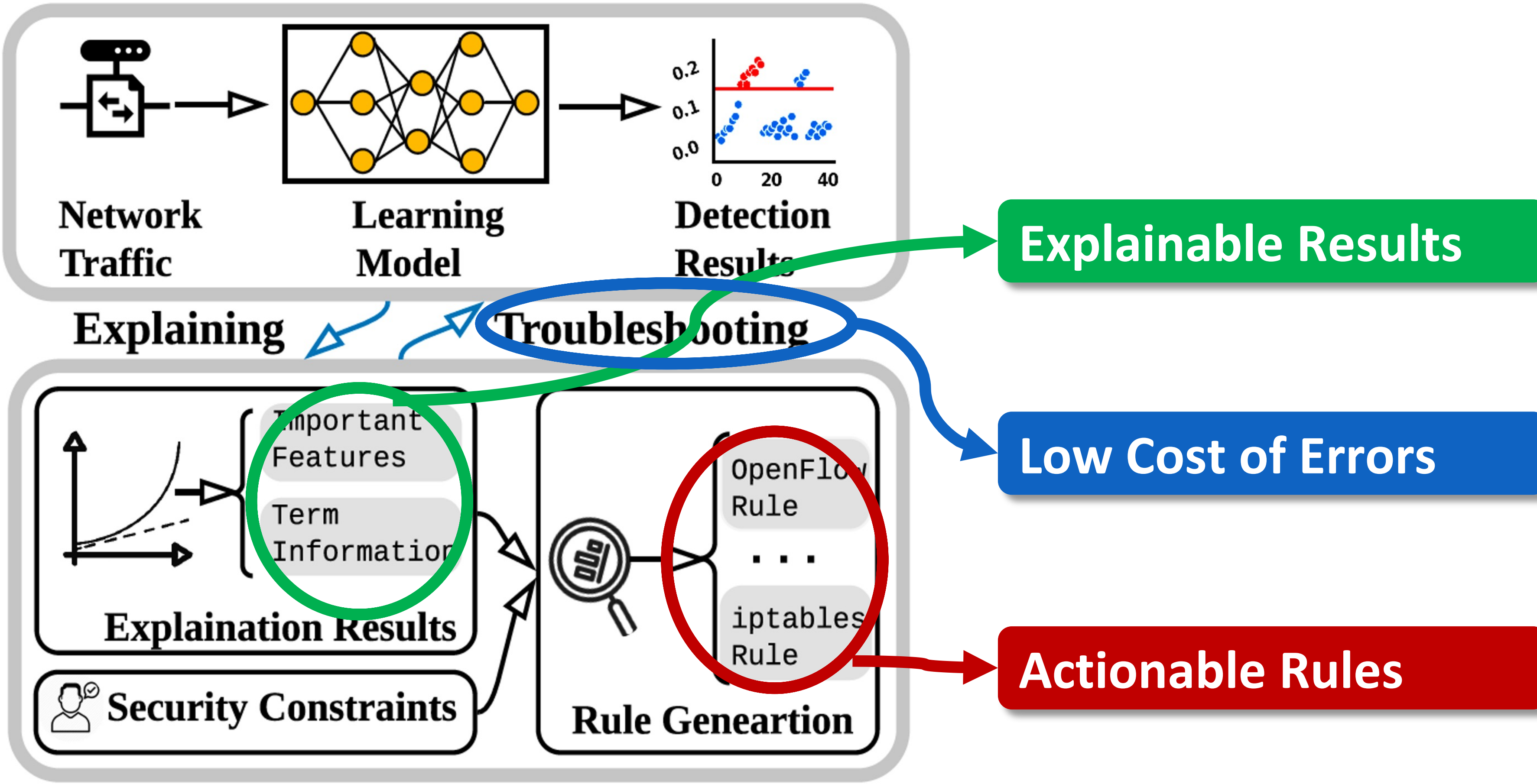
Different format levels of rule granularity

<nw_src = 192.168.1.10, tcp, tcp.syn, actions = drop, priority = 1, hard_timeourt = 60>

IPTABLES

< iptables -A INPUT -i eth0 -p tcp --tcp-flags SYN -s 192.168.1.10 -j DROP>

xNIDS: explaining deep learning-based NIDS for active intrusion response



Explaining DL-NIDS detection results

Approximating History

$$\underset{g}{argmin}\Big\{\underbrace{\mathcal{L}(f,g)}_{\text{Fidelity}}+\underbrace{\lambda\cdot\phi(\beta)}_{\text{Sparsity}}\Big\}\quad s.t.\ \Big\{\underbrace{||f(\mathbf{x}_t,\mathbf{X}'_{t,m})-y_t||_1<\delta}_{\text{History inputs}}\Big\}$$

Weighted Sampling

$$\mathbf{Z}_{t,m}\sim\mathcal{W}(\mathbf{X}'_{t,m},\mathbf{p})\quad \mathbf{p}\in(0,1)^m$$

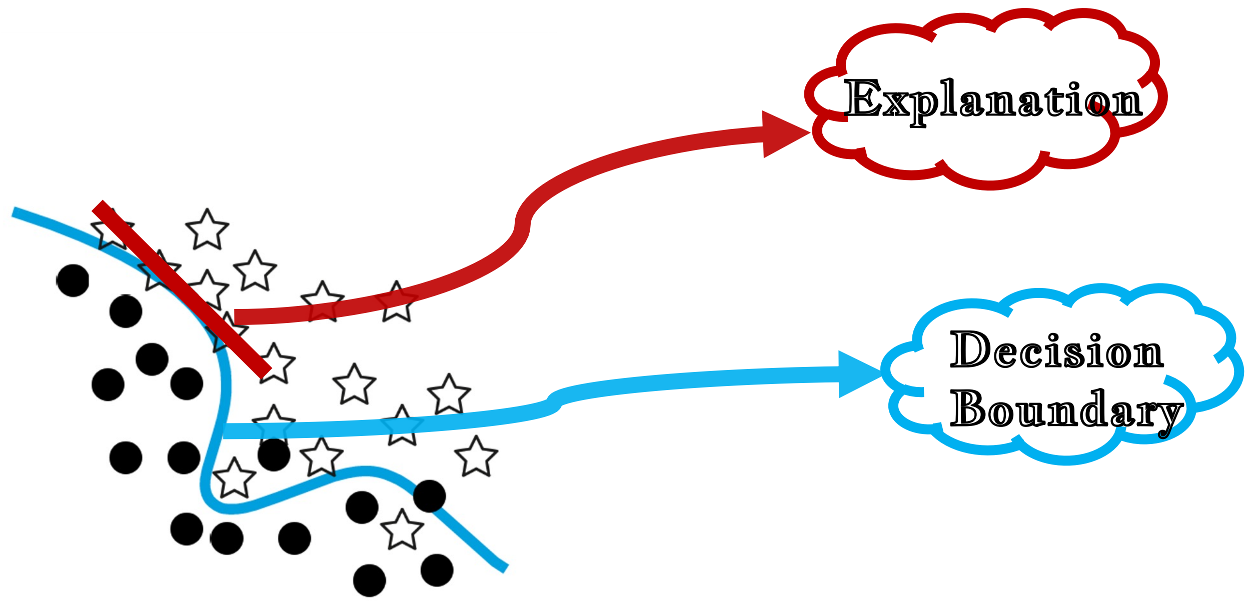
Capture Dependencies

$$\mathbf{x}_t^q=1_{A_q}:\mathbf{x}_t$$

$$\sum_{q=1}^{M}||\mathbf{x}_t^q||_1=||\mathbf{x}_t||_1\quad and\quad \mathbf{x}_t^i\cdot\mathbf{x}_t^j=0\ (i\neq j)$$

$$1_{A_q}:\mathbf{x}_t\rightarrow\{0,1\},\quad j\mapsto\begin{cases}1, j\in A_q\\0, j\notin A_q\end{cases}$$

Explaining DL-NIDS detection results

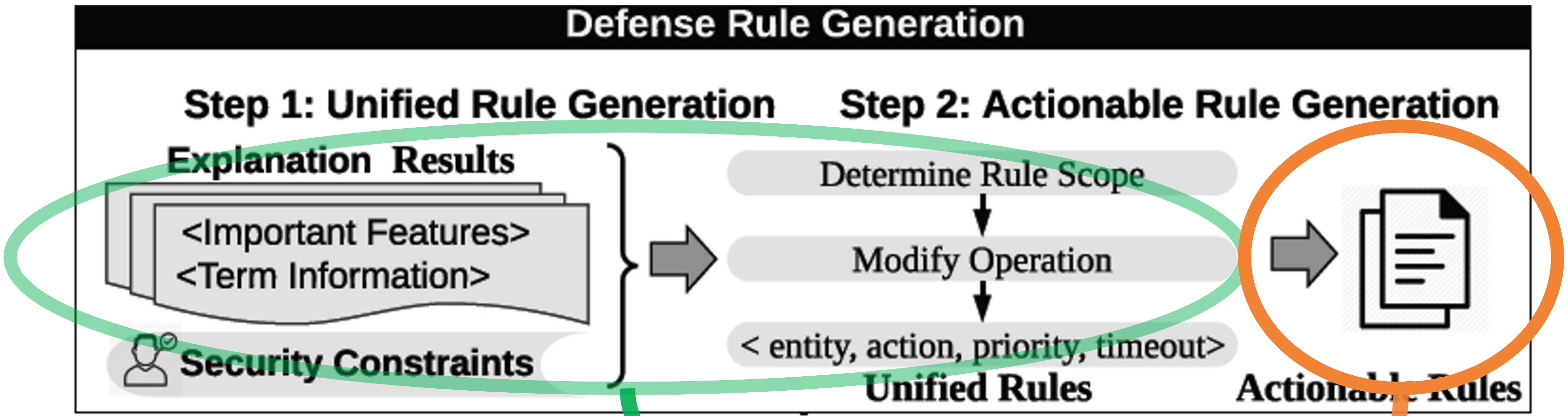


$$\underset{\beta}{argmin}\left\{ ||f-g||_2^2 + \underbrace{(1-\alpha)\lambda\sqrt{p_q}\sum_{q=1}^{Q}||\beta_q||_2}_{\text{Group Sparsity}} + \underbrace{\alpha\lambda||\beta||_1}_{\text{Feature Sparsity}} \right\}$$

Sparse on cross-group level

Sparse on intra-group level

Defense Rule Generation



Step 1: Generating Unified Defense Rule

Step 2: Generating Actionable Rules

Defense Rule Scope

Per-flow scope

Per-host scope

Multi-host scope

Algorithm 1: Determine Defense Rule Scope

Input: Explanation Result (T, F); /* T is term information, F are important features */

Output: *scope* /* Defense Rule Scope */

max = *Max* ($T.IP$, $T.MAC$, $T.protocol$, $T.port$)

if *max* == $T.protocol$ *or* $T.port$ **then**

    scope = *multi-hosts*;

else

    if F *contains multiple protocols or ports* **then**

        scope = *per-host*;

    else

        scope = *per-flow*;

return *scope*;

Unified Defense Rules

```
Notation: Integer n, Wildcard *

Entity          entity      ::= < IP, MAC, port, protocol, flag>
                IP          ::= < src_IP, dst_IP >
                MAC         ::= < src_MAC, dst_MAC >
                port        ::= < src_port, dst_port >
                protocol    ::= tcp | udp | icmp | arp | http | *
                flags       ::= tcp.syn | tcp.ack | tcp.fin | *

Action          action      ::= drop | allow | modify | whitelist
Priority        priority    ::= n
Timeout         timeout     ::= n
Unified Rule    rule        ::= < entity, action, priority, timeout >
```

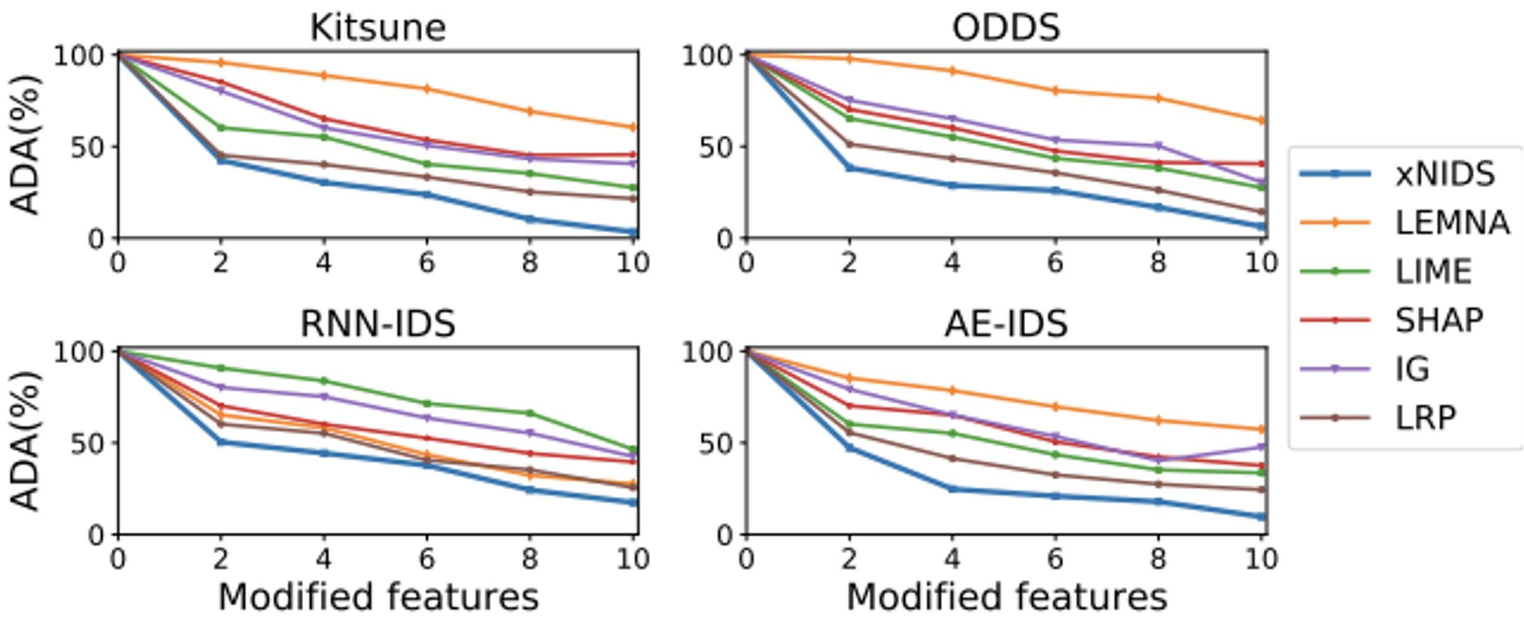
Evaluation

- Fidelity, Sparsity, Completeness and Stability of Explanation
- Practicability and Efficiency of Defense Rules
- Showcasing Troubleshoot and Active Response

Fidelity of Explanation

System	Kitsune	ODDS	RNN-IDS	AE-IDS
LIME	0.509	0.531	0.770	0.521
SHAP	0.643	0.578	0.593	0.593
LEMNA	0.830	0.856	0.525	0.748
IG	0.608	0.618	0.690	0.623
LRP	0.409	0.427	0.507	0.438
xNIDS	**0.316**	**0.325**	**0.430**	**0.331**

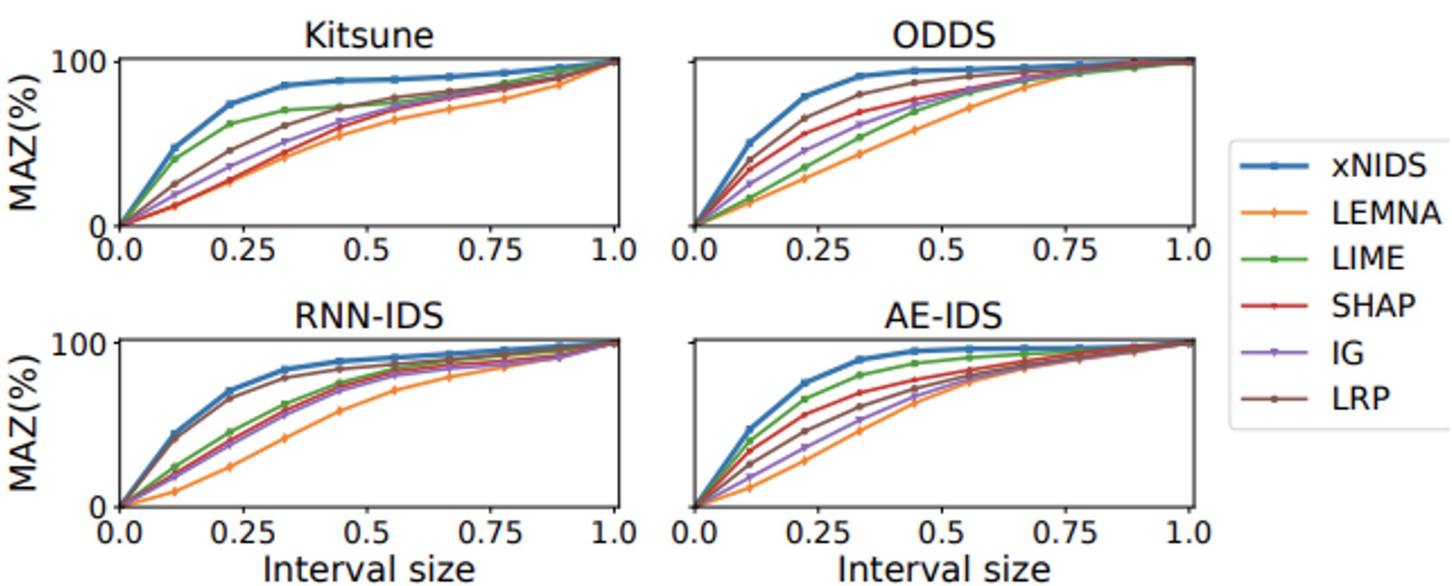
Fidelity: examine how faithful the explanation method captures the important features



Sparsity of Explanation

System	Kitsune	ODDS	RNN-IDS	AE-IDS
LIME	0.650	0.762	0.745	0.667
SHAP	0.685	0.647	0.680	0.760
LEMNA	0.542	0.599	0.569	0.604
IG	0.577	0.713	0.637	0.632
LRP	0.605	0.680	0.655	0.708
xNIDS	**0.774**	**0.814**	**0.775**	**0.806**

Sparsity: how sparse the selected important features are



Overall Comparison

Criteria	LIME	SHAP	LEMNA	IG	LRP	xNIDS
Fidelity	◐	◐	○	◐	◐	●
Sparsity	◐	◐	○	○	○	●
Completeness	○	○	○	◐	◐	●
Stability	○	○	○	●	●	◐
Rule Generation	/	/	/	/	/	●

Completeness: an explanation is complete if it can create proper results for all possible input samples

Stability: examine whether the explanation is stable among multiple runs

Practicability of Rule Generation

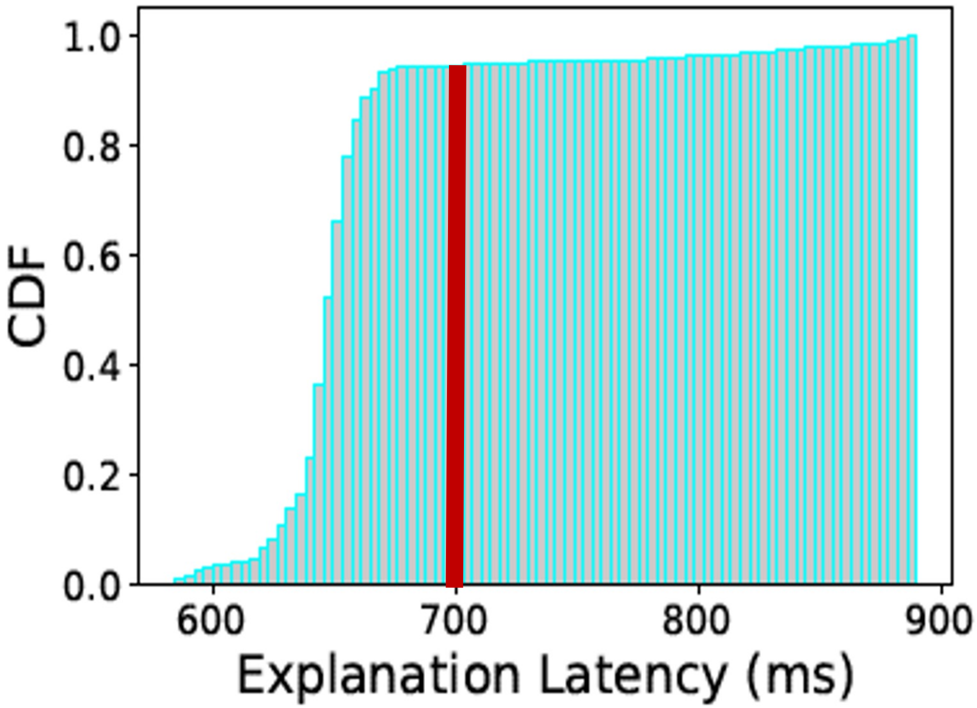
Defense Tool	Entity	Action	Priority	Timeout
OpenFlow	●	●	●	●
iptables	●	●	◐	◐
Pfsense	◐	●	◐	◐
Squid	◐	●	◐	◐

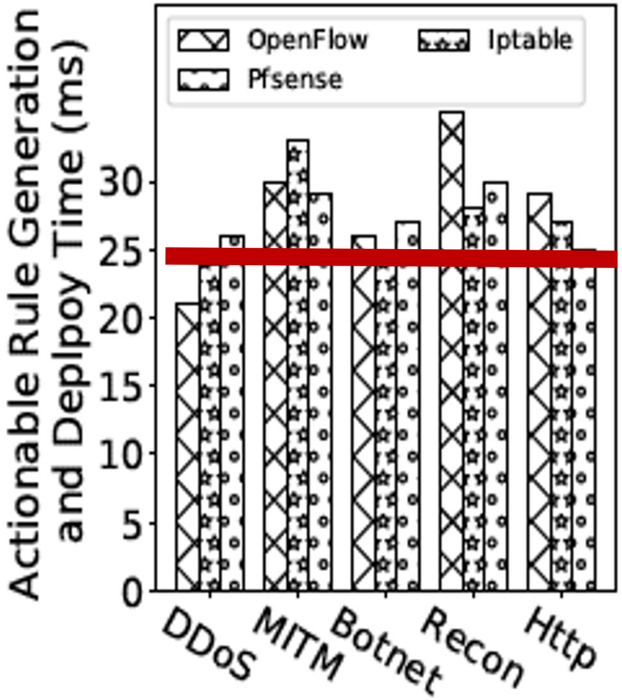
```
R1:<entity(src_ip = 157.240.1.9, dst_ip = 157.240.1.3,
TCP,  TCP_flags=syn),
actions = drop,  priority = 1,  timeout = 6000>
```

```
R2:<entity(src_ip = 157.240.1.12,
src_mac = dc:a9:04:bc:7e:42 )
action = drop, priority = 3, timeout = MAX>
```

```
R3:<entity(src_ip=*, dst_port = 1900 )
action = drop, priority = 4, timeout = MAX>
R4:<entity(src_ip=157.240.1.13, dst_port = 1900 )
action = allow, priority = 3, timeout = MAX>
```

Efficiency of Rule Generation

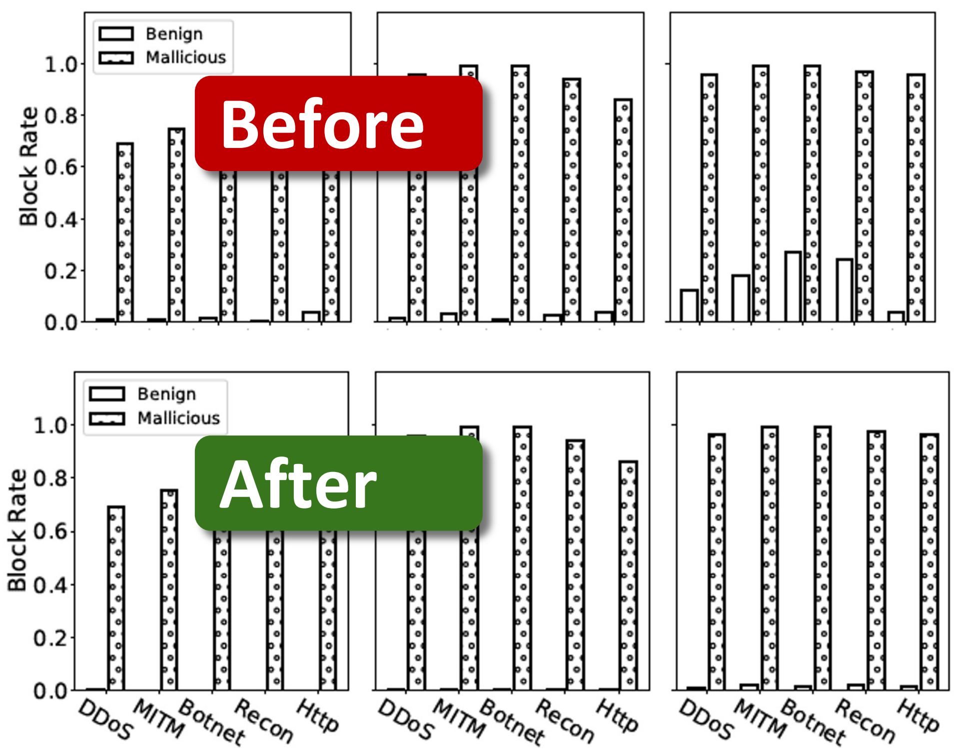




Efficiency: 95% of the explanation latency is under 700ms, while average latency for generating actionable rule is 25ms

Troubleshooting and Active Response

Error Type	Before		After		Reducing Rate
	FP	Blocked	FP	Blocked	
Type-1	137583	136425	137583	0	100%
Type-2	45744	44371	16012	15369	65.36%
Type-3	35676	35562	1192	1141	96.79%



Troubleshooting: xNIDs can reduce error cost case by case

Active response: after troubleshooting xNIDS can precisely block the malicious traffic

Conclusion and Future Work

- xNIDS:
  - Explain the detection results of DL-NIDS
  - Generate defense rules for active responses
- Future work
  - Adopt the transformer model to re-design DL-NIDS and the attention mechanism to explain DL-NIDS for active response
  - Investigate how to improve the robustness and accuracy of DL-NIDS at the same time

fengwei@buffalo.edu

University at Buffalo