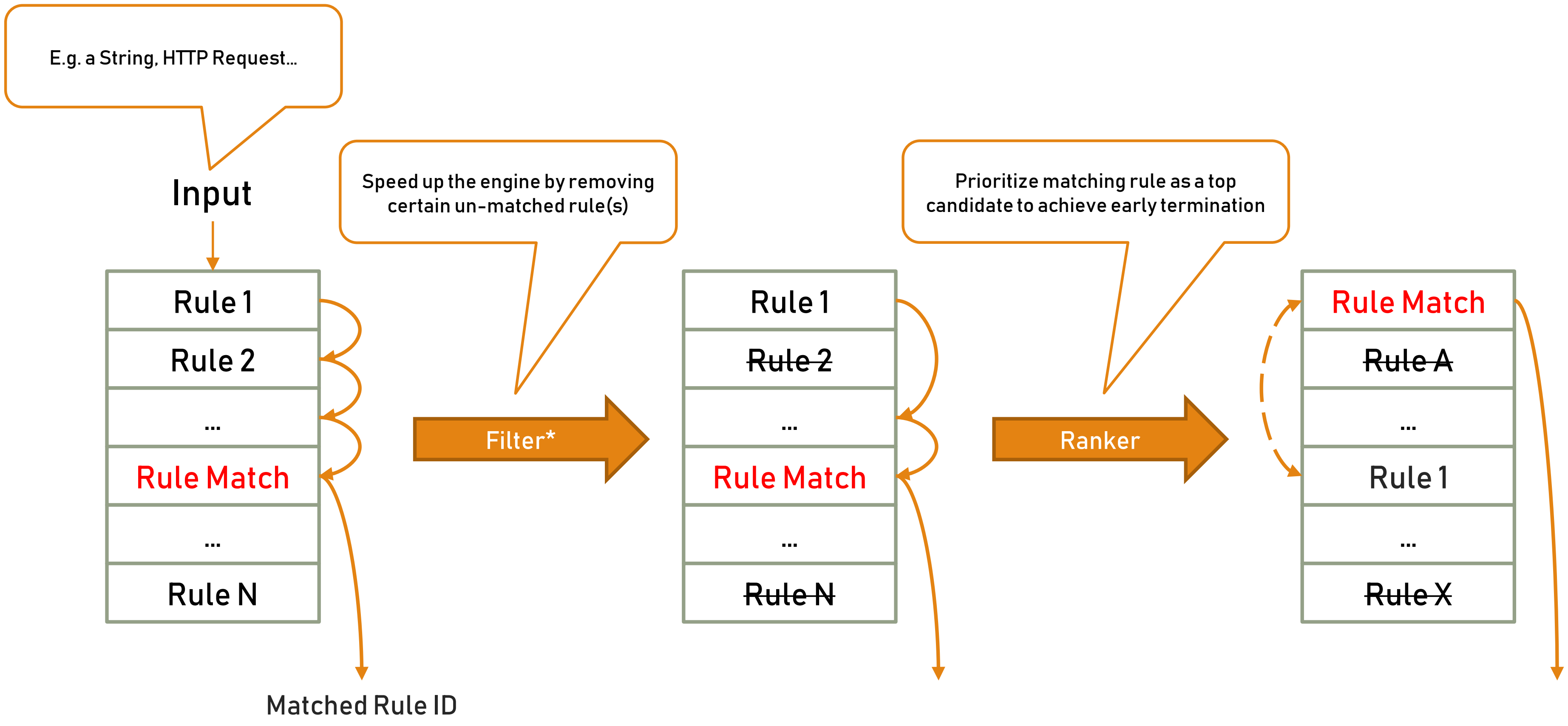
Accelerating Rule-matching Systems with Learned Ranker

ZHAO LUCIS LI	UNIVERSITY OF SCIENCE AND TECHNOLOGY CHINA
CHIEH-JAN MIKE LIANG	MICROSOFT RESEARCH
WEI BAI	MICROSOFT RESEARCH
QIMING ZHENG	SHANGHAI JIAO TONG UNIVERSITY
YONGQIANG XIONG	MICROSOFT RESEARCH
GUANGZHONG SUN	UNIVERSITY OF SCIENCE AND TECHNOLOGY CHINA

USENIX ATC '19
2019 USENIX Annual Technical Conference

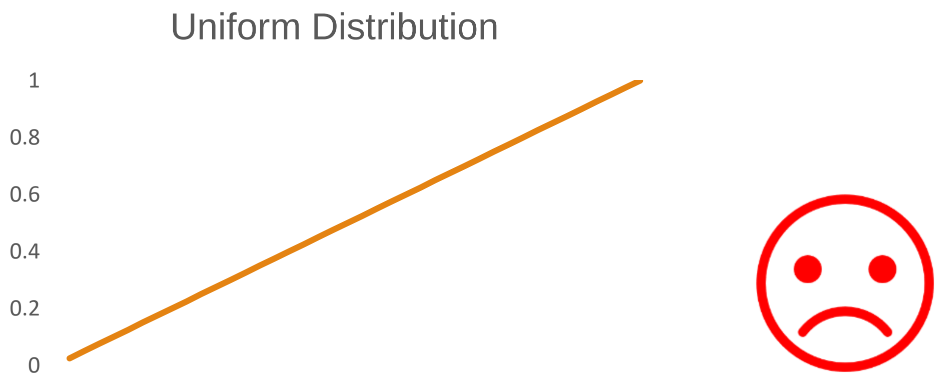
Rule Engine Matching Process



* Roesch, Martin. "Snort: Lightweight intrusion detection for networks."

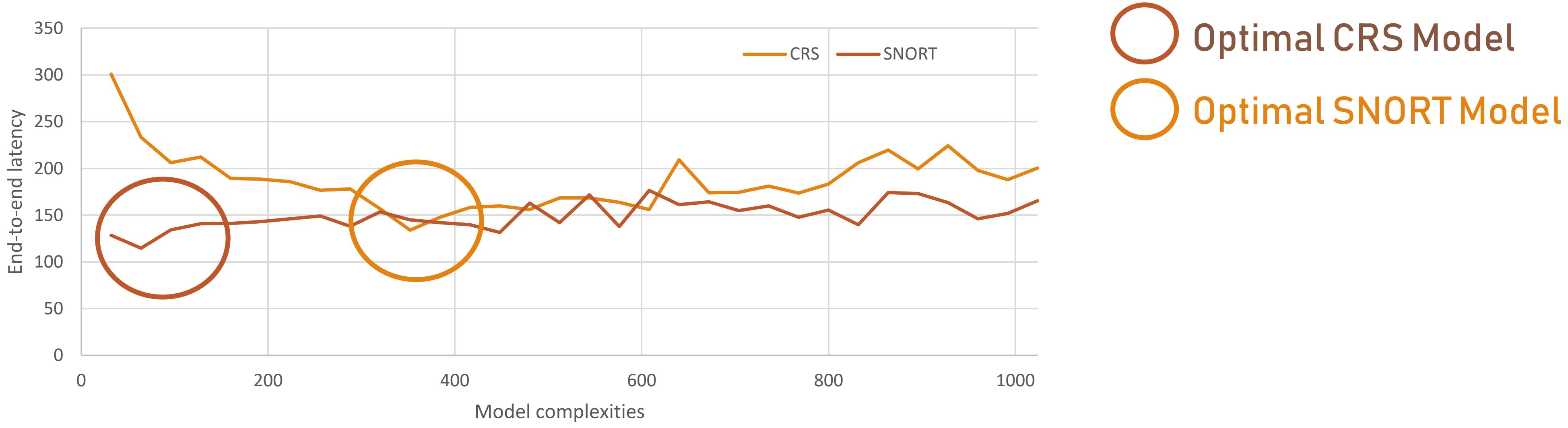
Challenges of Ranker Design

- Ranker should estimate input features, instead of assuming data stream distribution.
  - LRU or LFU based ranker orders ruleset for current input from historical data stream.
  - LRU or LFU is good at long-tailed data stream but bad in uniform distribution.



Challenges of Ranker Design

➢Learned ranker should consider the trade-off between inference cost and accuracy.



➢Learned ranker should consider training data quality.
  - ➢Artificial datasets might not provide sufficient insights to learn decision boundaries.
  - ➢Logged real-world system workloads might not cover all cases.

Performance Gains from Learned Ranker

- Average reduction in the number of rules that the rule engine needs to process.

Rule set	No rule ranker	Rule ranker	Reduction
CRS	22.38	1.68	92.49%
SNORT	91.56	1.34	98.54%

- Average reduction in latency for matching one input on different rule engines for CRS.

Rule engine(regex)	No rule ranker	Rule ranker	Reduction
PCRE	1878.79 μsec	404.36 μsec	78.47%
PCRE with JIT	773.82 μsec	185.65 μsec	78.81%
RE2	206.01 μsec	55.15 μsec	73.22%

Thanks for watching!

USENIX 2019, July 12th 11: 50 am–1:10 pm

Track II : Machine Learning Applications & System Aspects