Re-architecting Traffic Analysis with *Neural* Network Interface Cards

Giuseppe Siracusano, Salvator Galea, Davide Sanvito, Mohammad Malekzadeh,
Gianni Antichi, Paolo Costa, Hamed Haddadi, Roberto Bifulco

Joint work of

NEC

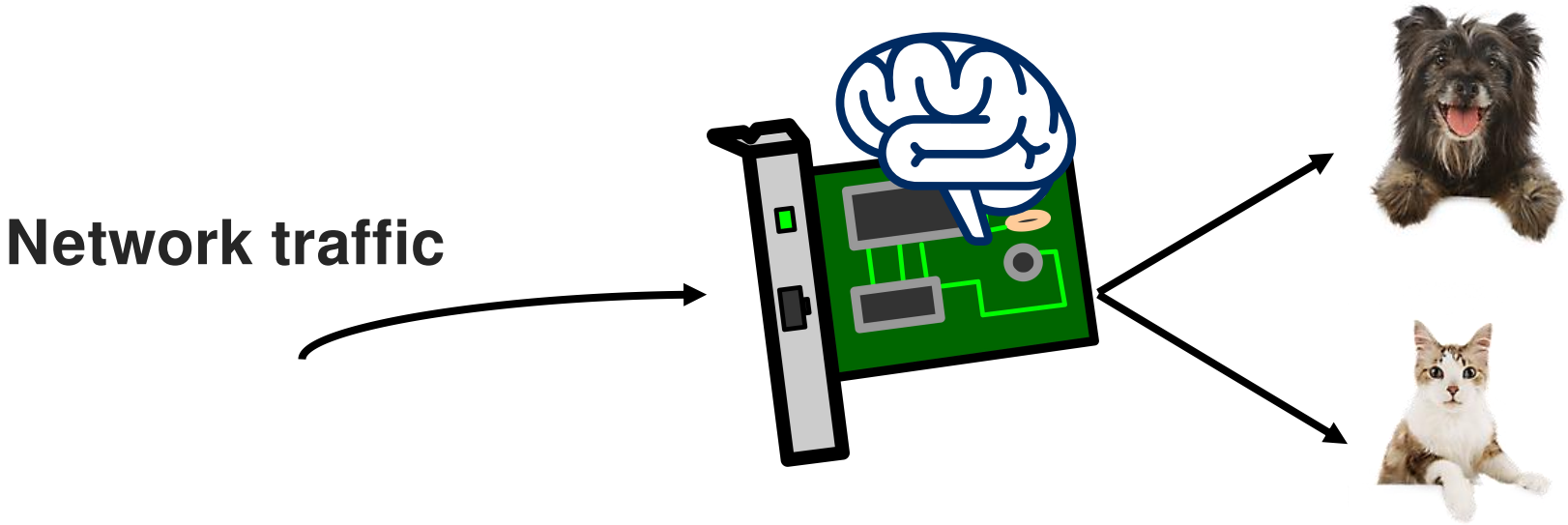
UNIVERSITY OF CAMBRIDGE

Imperial College London

Queen Mary University of London

Microsoft Research

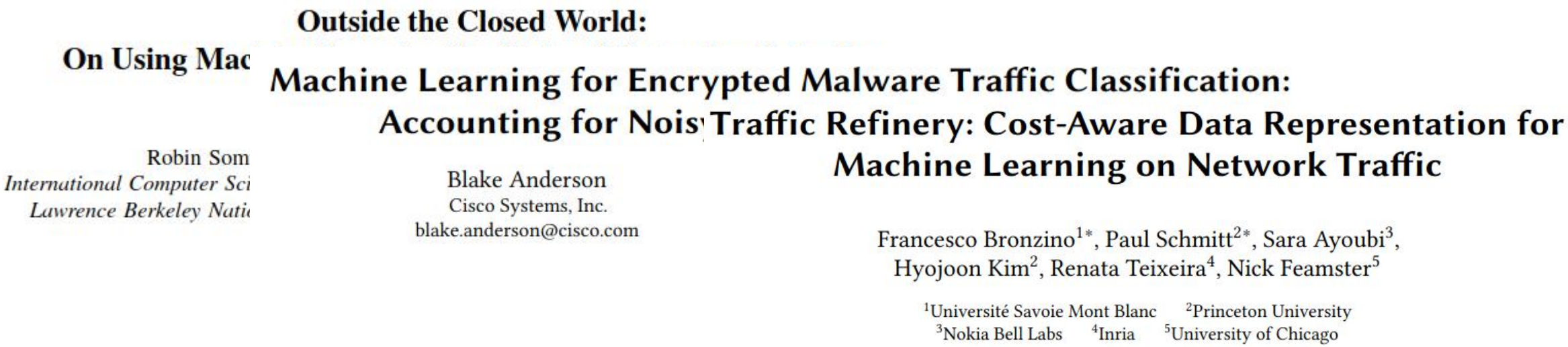
Network Interface Cards and Neural Networks



Why?

Online Traffic Analysis

- Fundamental building block in today's networks
- Drivers for adoption of traffic analysis based on Machine-Learning
  - Complexity of network traffic patterns
  - Use of encrypted communications



On Using Mac

Robin Som
International Computer Sci
Lawrence Berkeley Nati

Outside the Closed World:

Machine Learning for Encrypted Malware Traffic Classification:
Accounting for Nois

Blake Anderson
Cisco Systems, Inc.
blake.anderson@cisco.com

Traffic Refinery: Cost-Aware Data Representation for Machine Learning on Network Traffic

Francesco Bronzino[1]*, Paul Schmitt[2]*, Sara Ayoubi[3],
Hyojoon Kim[2], Renata Teixeira[4], Nick Feamster[5]

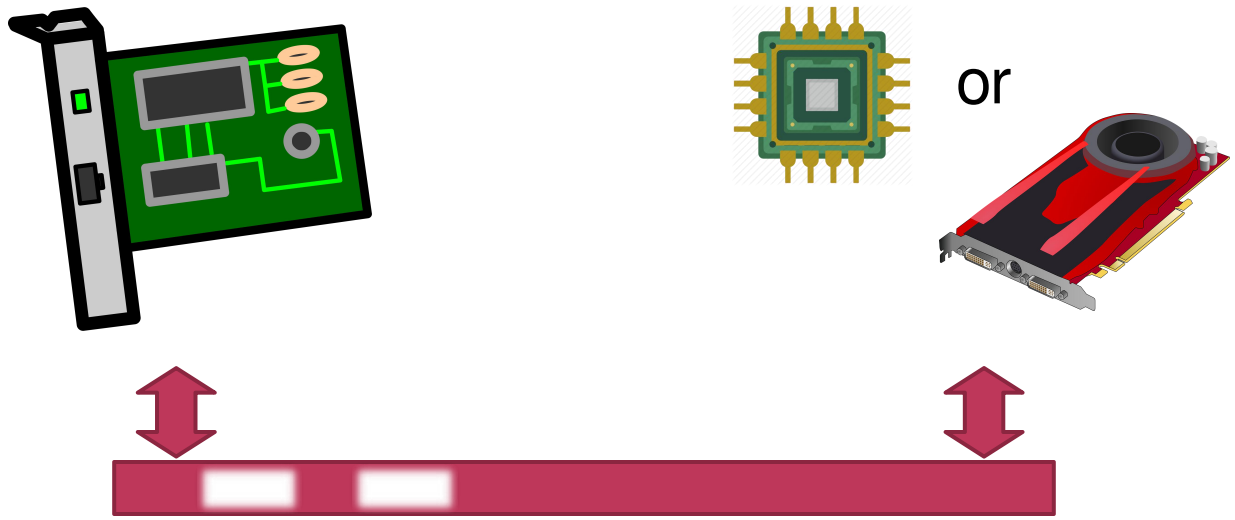
[1]Université Savoie Mont Blanc [2]Princeton University
[3]Nokia Bell Labs [4]Inria [5]University of Chicago

- **Challenging throughput and latency requirements**

ML based traffic analysis on dedicated executor

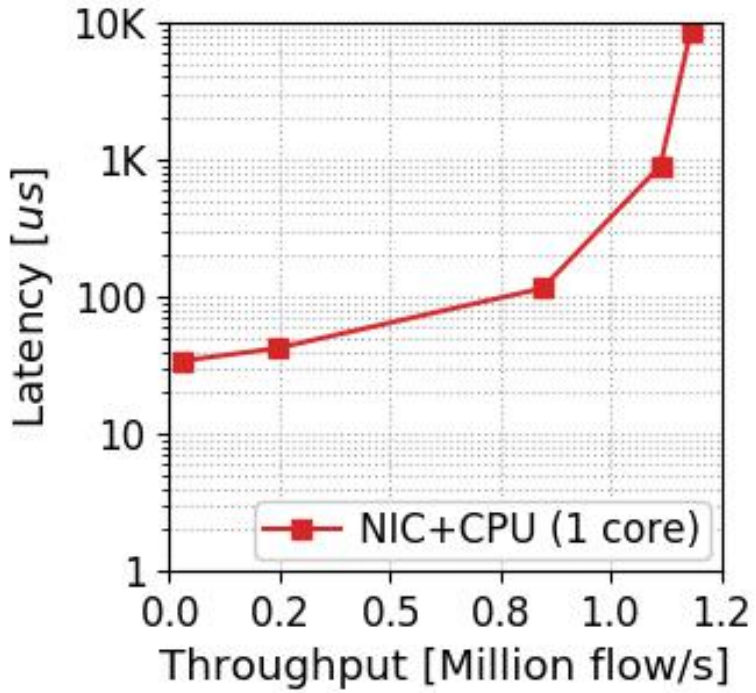
Flow statistic collected in the data plane

ML inference in a separate executor



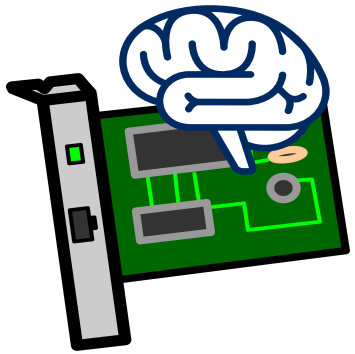
Why is this a problem?

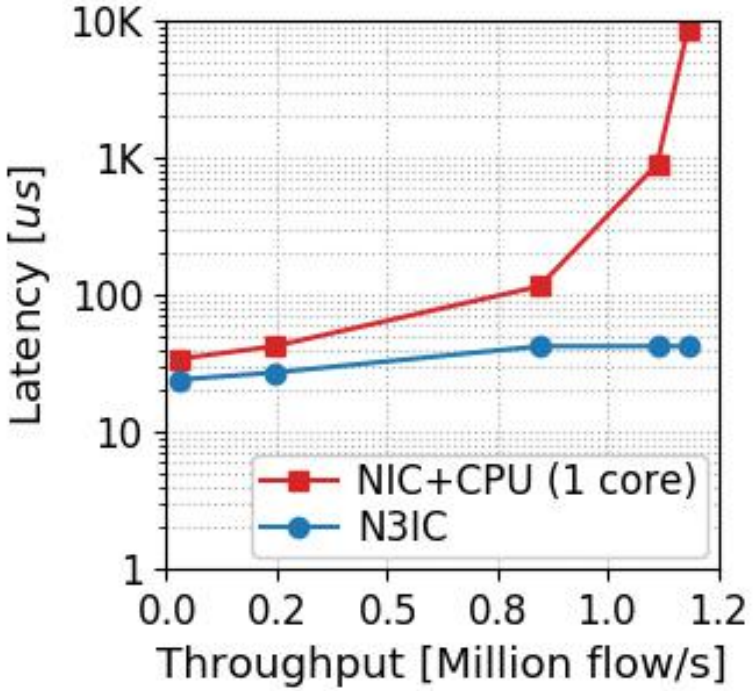
Data movement is the bottleneck



- Simple experiment
  - Feature extraction in the data plane, processing in a dedicated executor

What if we offload analysis to the NIC?





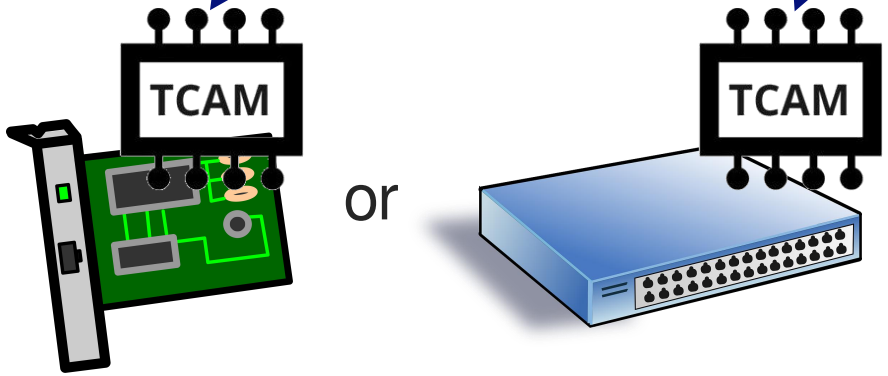
Look at the blue line

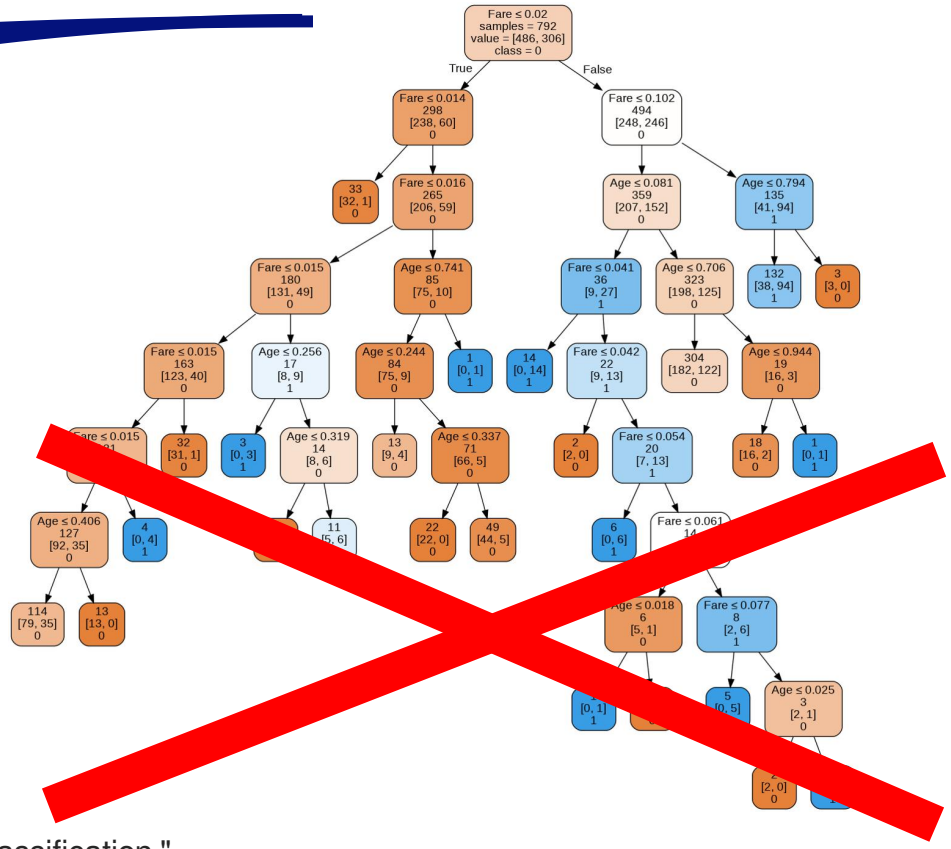
More benefits: save CPU/GPU processing

We are not alone...

TCAM is expensive

Only small models





- State-of-the-art solutions [1, 2] implement analysis with widely used ML techniques

[1] Xiong, Zhaoqi, and Noa Zilberman. "Do switches dream of machine learning? toward in-network classification."
[2] Coralie Busse-Grawitz, Roland Meier, Alexander Dietmüller, Tobias Bühler, and Laurent Vanbever.
pfoest: In-network inference with random forests.

Our Goal

- ML based traffic analysis on commodity SmartNICs
  1. Efficiently leverages programmable NICs' hardware
  2. Accuracy comparable to existing ML-based traffic analysis software solutions
  3. Achieve high throughput and low latency

- **Leverage NIC architecture parallelism**

Challenges

1. Highly parallelizable algorithm

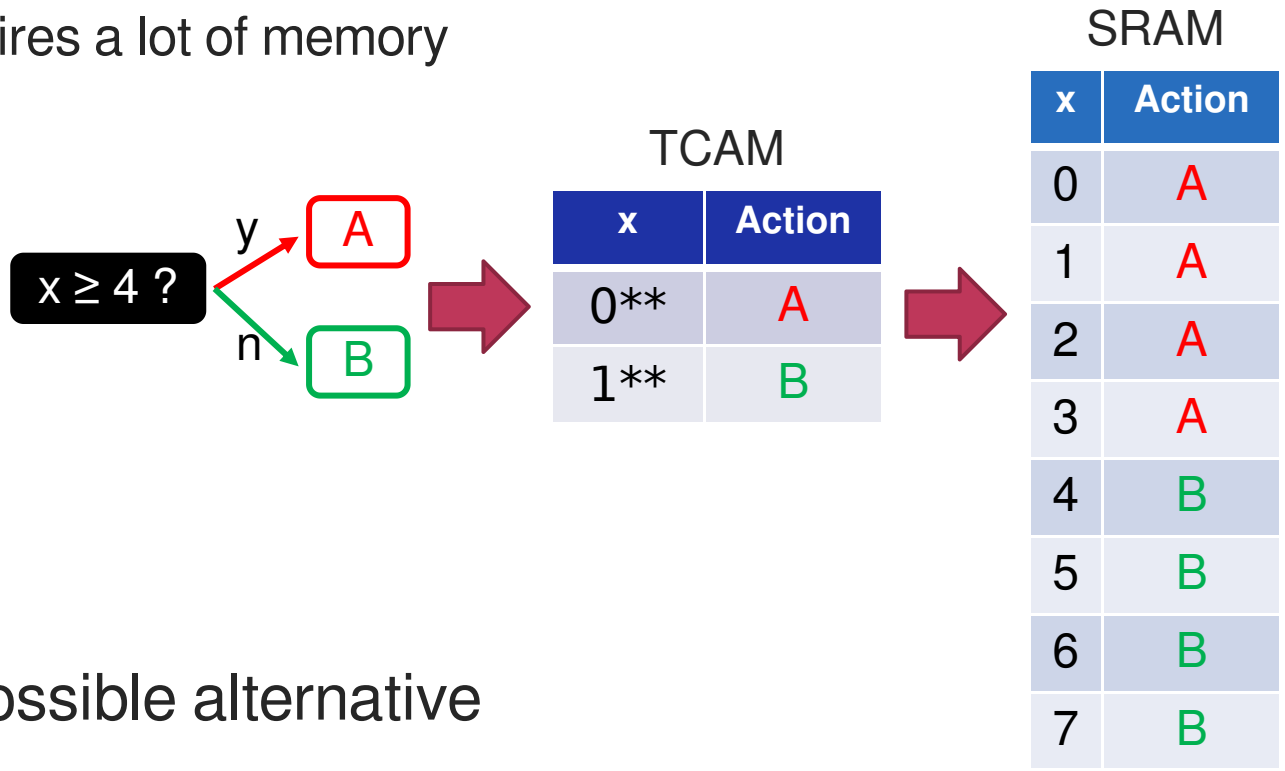
2. Limited amount of fast on-chip SRAM
   - used to store forwarding/policy tables
   - little space available for application data

3. Missing complex arithmetic functions
   - i.e., multiplications or floating-point operations

- **The key is to exploit the right ML tool for the job**

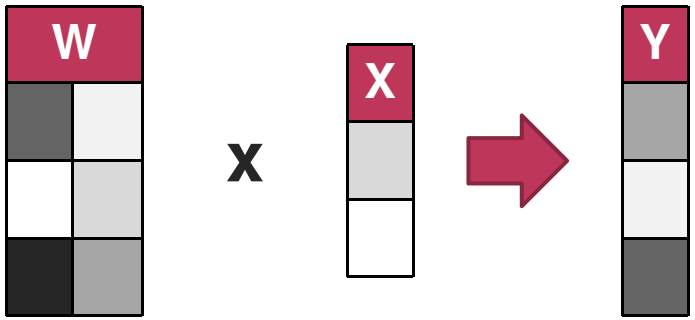
Current Traffic Analysis tools

- Decision Tree or Random Forest
  - Ternary matching required for data plane implementation
  - SRAM implementation requires a lot of memory



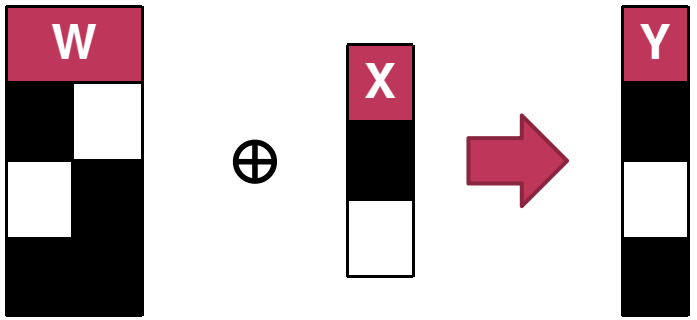
- Neural Networks are a possible alternative

Neural Networks ?



- Parallelizable

Binary Neural Networks[1]

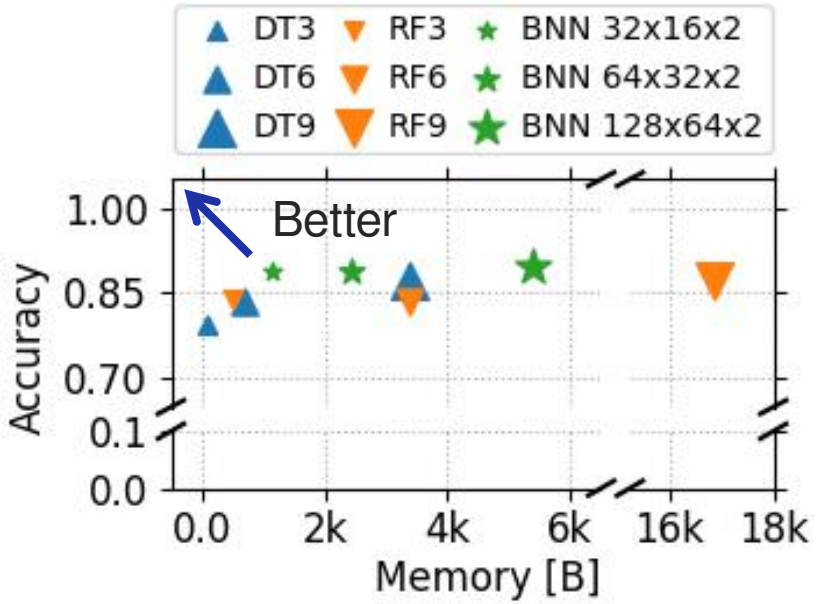


- Parallelizable
- Reduced memory footprint
- Operations supported by most hardware platforms

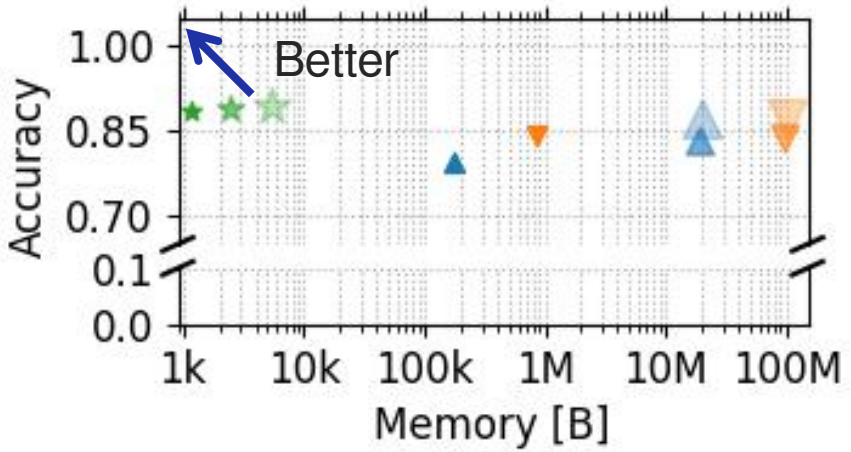
- **Single bit** representation instead of **8-32 bit** floating-point
- **XOR**, **popcnt** and **sign** instead of sum, **multiplication** and **non linear activation** function

[1] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks.

Classification w/ BNNs



DT/RT in TCAM



DT/RT in SRAM

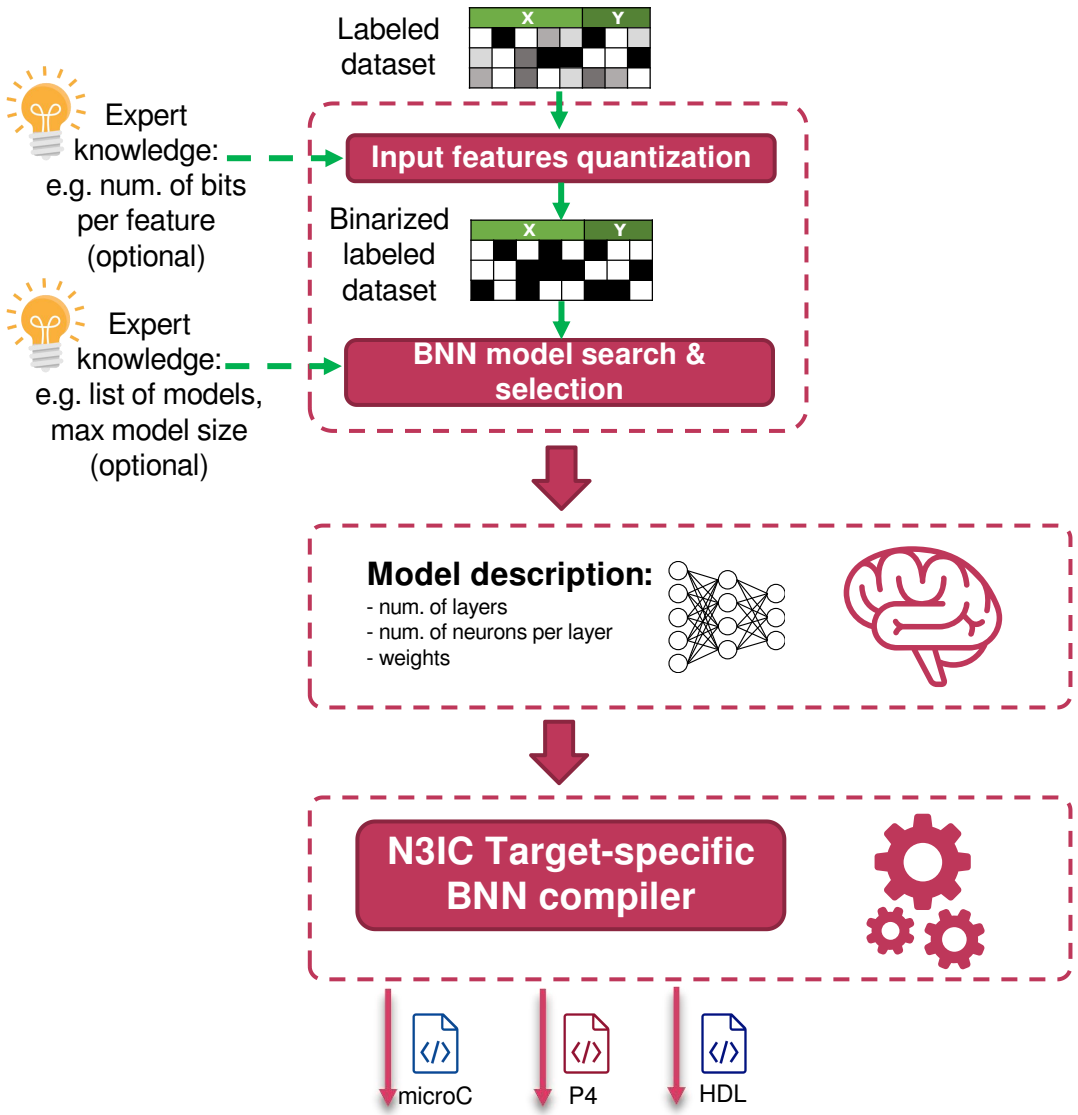
- Use case
  - Security Anomaly Detection
- Legend
  - DT = Decision Tree
  - RT = Random Forest
  - BNN = Binary Neural Network

Better/Similar accuracy and less memory consumption

BNNs can replace DT and RF for traffic analysis.

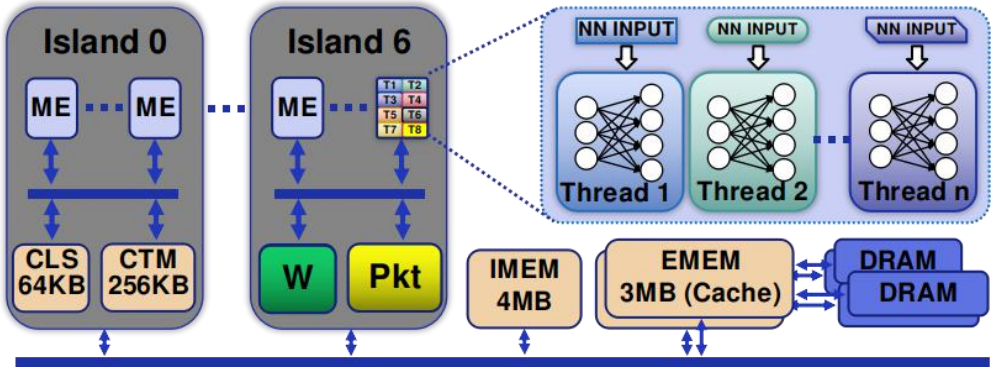
How do we run them in a NIC?

Neural Networks on the NIC (N3IC)

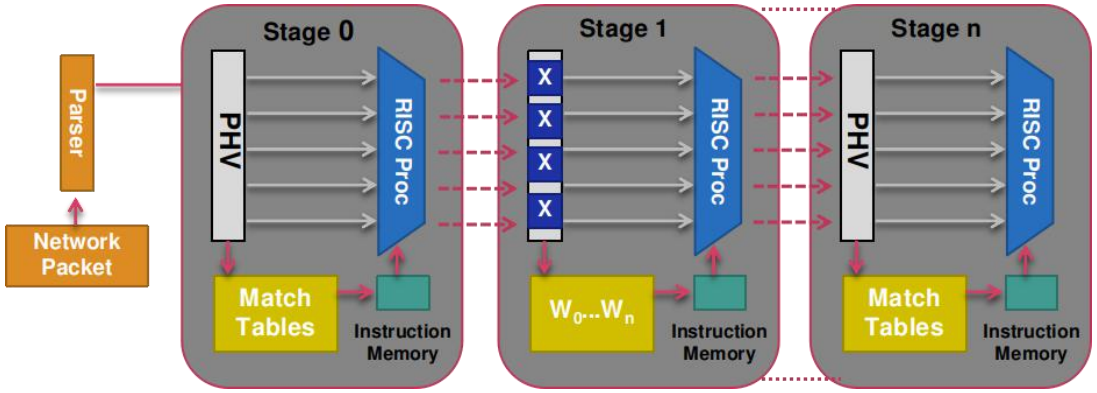


- Trains a BNN using a labeled dataset provided by the user
- Compiles BNN models into target-specific executables

N3IC hardware targets

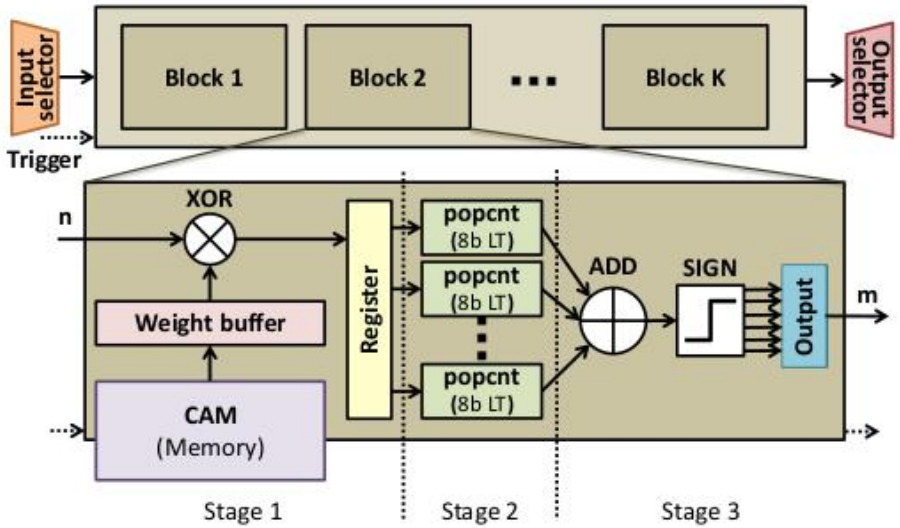


- Netronome NFP
  - thread level parallelism
  - weights are stored in CLS (SRAM)



- PISA
  - pipeline-level parallelism
  - weights are stored in exact MAU

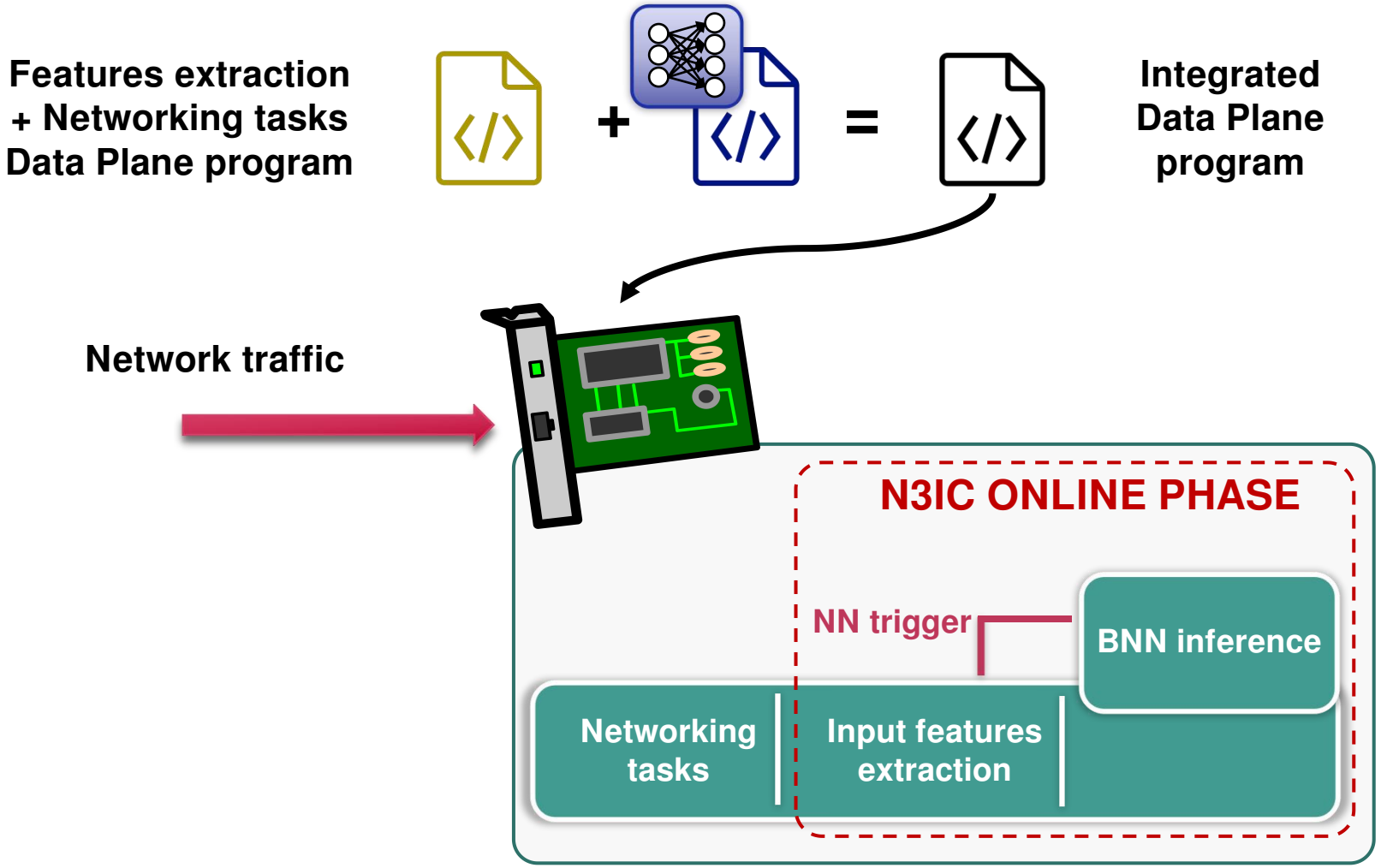
One step further



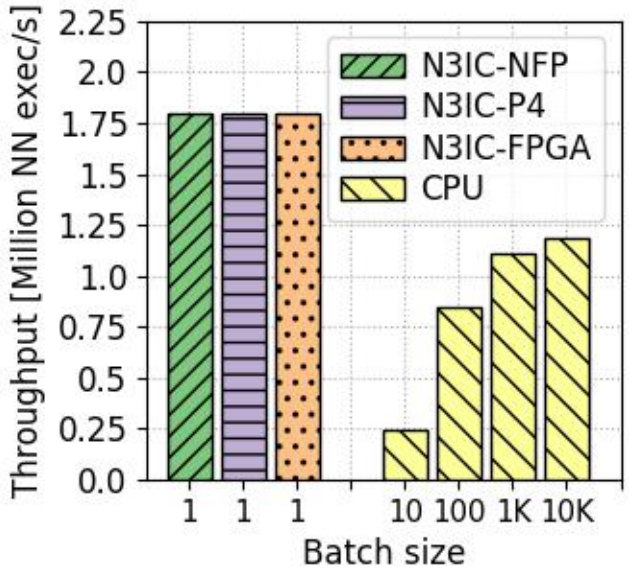
Design	LUT	BRAM
Reference NIC (RN)	11.40%	13.20%
RN + simple Feature Extraction (FE)	11.56%	17.60%
RN + simple FE + **N3IC-FGPA**	12.16%	18.80%
RN + advanced FE	21.56%	32.60%
RN + advanced FE + **N3IC-FPGA**	22.86%	33.80%

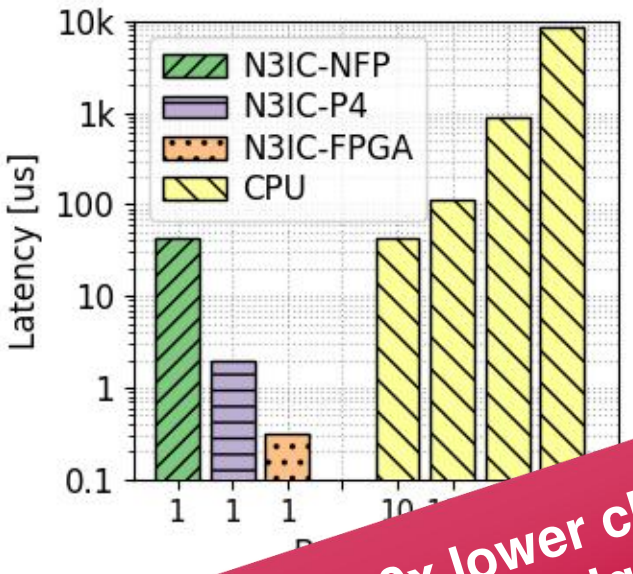
- Native Hardware support for BNNs
- Prototype on the NetFPGA using RTL description language
- Only needs a modest 1-2% of a Xilinx Virtex7 FPGA's logic resources.

Putting things together



Does it work?

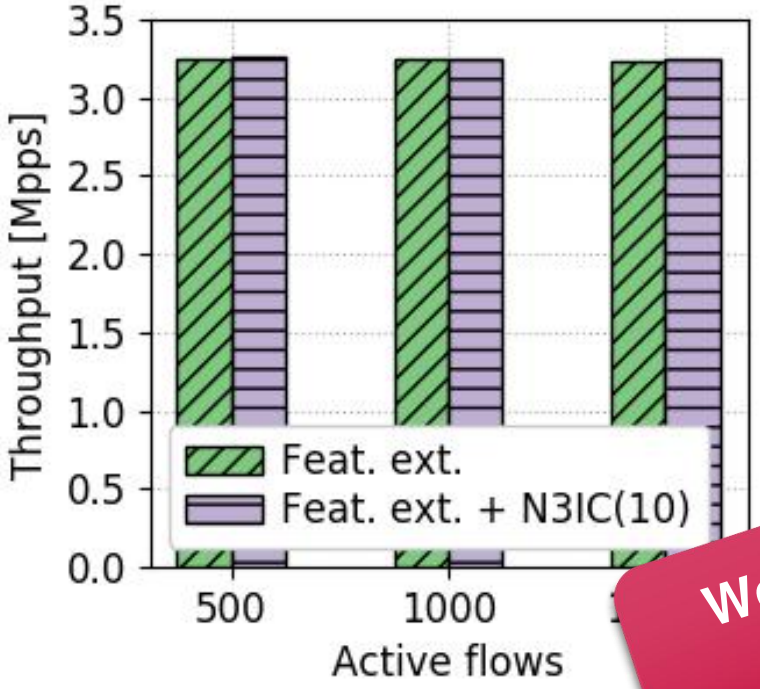




Up to a 100x lower classification latency, and 1.5-7x higher throughput

- N3IC vs BNN CPU executor
  - feature extraction is always performed in the NIC
- Traffic input
  - 1.8M flows per second
  - 10 packets per flow at 40Gb/s w/ 256B pkt size
- HW targets
  - N3IC-NPF = Netronome SmartNIC
  - N3IC-P4 = P4 to FPGA
  - N3IC-FPGA = native FPGA
  - CPU executor

Does it work?



Workloads can be efficiently co-located.

- Data plane app
  - Feature extraction and full TCP tracking w/o and w/ N3IC-NFP
- Input
  - 40Gb/s distributed among 500, 1k, and 10k TCP parallel flows.
  - packet size ~1.5KB

Summary

- BNNs can replace widely-adopted DTs and RFs for traffic analysis use cases
- They can be efficiently implemented in different architectures
- N3IC compiles BNN models into implementations that can be directly integrated in the data plane of SmartNICs
- Adding BNN Dedicated HW in SmartNIC is cheap
  - Enabler for other use cases (see the paper)

Limitations

- Only features that can be computed/extracted within the data plane can be used as input
- Limited to small models

Follow up questions

giuseppe.siracusano@neclab.eu

code

https://github.com/nec-research/n3ic-nsdi22

Thank you!