Lessons in Practical In-Network Classification

Changgang Zheng, University of Oxford

Joint work with Mingyuan Zang (DTU), Xinpeng Hong, Riyad Bensoussane (Oxford), Zhaoqi Xiong, Thanh T. Bui, Siim Kaupmees (Cambridge), Antoine Bernabeu (École Centrale de Nantes), Lars Dittmann (DTU), Stefan Zohren (Oxford), Shay Vargaftik, Yaniv Ben-Itzhak (VMWare) and Noa Zilberman (Oxford)

IETF-115, November 2022
Do Switches Dream of Machine Learning?

HotNets 2019

- Mapping trained ML models to programmable network devices
- Supporting Decision trees, SVM, K-means, Naïve Bayes
- Using a baseline P4 design per algorithm
- Classification rules = tables updates
- Running on NetFPGA-SUME and BMv2
Automating In-Network Machine Learning

2022

• End-to-end automated process
  • Training
  • P4 generation
  • P4runtime and table rules generation
  • Test and validation
  • Running on hardware

• Models: >12 models (e.g., XGBoost, Isolation Forest, PCA, KNN, Autoencoder, …)
• Targets: Tofino, Tofino2, P4Pi T4P4S+BMv2, Bmv2, NVIDIA (WIP), FPGA (WIP)
In-Network Classification

How hard can it be?
Challenge: Limited Number of Stages

Solution: Parallelization

(a) SwitchTree & pForest

(b) Ily & Planter

Zheng et al, Automating In-Network Machine Learning, 2022
Challenge: Limited Memory

Solution: Different mappings, LPM tables + smart drop

Zheng et al, Automating In-Network Machine Learning, 2022
Challenge: Maintaining Network Functionality

Solution: Parallelism + Resource Efficiency

- Integrated with Intel’s switch.p4
- A large model consumes 5%-65% resources relative to switch.p4
- Model & use-case dependent
Challenge: Limited Resources (combined)

Solution: Trade offs + Use Case Dependent

Zheng et al, Ilisy: Practical In-Network Classification, 2022
Challenge: Runtime Retraining & Updates

Solution: Digest messages + Shadow updates

- P4Pir is running on P4Pi
- In-network analysis for smart IoT Gateways
- Sending digest information to control plane
- Model retraining (on server)
- Shadow updates to table entries
- Hitless
Challenge: Network Performance

Solution: By Design

- Design fits commodity programmable switch ASIC
  - No recirculation / resubmission
  - No control plane dependencies
  - No special modules
- Achieves 100% line rate (64x100G Tofino)
- Sub-microsecond latency
  - Same or less than switch.p4
Challenge: Limited Model Size

Solution: Hybrid Model Deployment

- "Small" model on the switch
- "Large" model on the backend
- Use decision confidence to either classify or send to backend
Hybrid Model Deployment

Anomaly detection use case

HFT use case

Figure 7: Anomaly detection: fraction of traffic handled by the switch and misclassification rate.

Figure 11: Latency sensitive financial transactions: The effect of confidence threshold
Summary

• In-Network Classification is feasible*
  • Line rate
  • Commodity switches
  • Coexists with network functionality
  • Scalable / Hybrid deployment
• Use cases
  • Anomaly detection
  • IoT gateways
  • HFT
  • Ideas?

* It is not the ultimate solution to world’s problems!
Use Case Ideas? New Challenges?

Changgang Zheng
✉ changgang.zheng@eng.ox.ac.uk
https://changgang-zheng.github.io/Home-Page

Noa Zilberman
✉ noa.zilberman@eng.ox.ac.uk
https://eng.ox.ac.uk/computing

List of Papers:
Xiong & Zilberman, Do Switches Dream of Machine Learning?, 2019
Zheng et al, Planter: Seeding Trees Within Switches, 2021
Zheng et al, Ilsy: Practical In-Network Classification, 2022
Zheng et al, Automating In-Network Machine Learning, 2022
Hong et al, Linnet: Limit Order Books Within Switches, 2022
Zang et al, P4Pir: In-Network Analysis for Smart IoT Gateways, 2022

EB solution and example
LB solution and example
DM solution and example
Scalability evaluation

We acknowledge support from VMWare and equipment donations from Intel and NVIDIA
We thank Tom King, Jeremy Daniel and Radostin Stoyanov for help with experimental setup and its optimization