In-Network ML

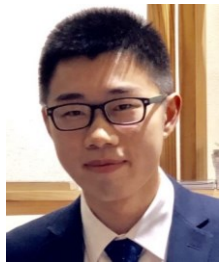
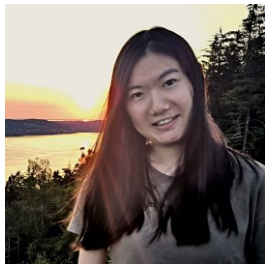
Inference at the Speed of Data

Noa Zilberman

noa.zilberman@eng.ox.ac.uk / planter@eng.ox.ac.uk

NANDA Workshop, September 2024

Acknowledgements


Many people have contributed to this research over the years:

Changgang Zheng, Mingyuan Zang, Xinpeng Hong, Zhaoqi Xiong, Riyad Bensoussane, Liam Perreault, Benjamin Rienecker, Peng Qian, Hongyi Chen, Damu Ding, Filippo Cugini, David Bowden, Kari Koskinen, Thanh T. Bui, Siim Kaupmees, Antoine Bernabeu, Tomasz Koziak, Lars Dittmann, Haoyue Tang, Aosong Feng, Leandros Tassiulas, Stefan Zohren , Shay Vargaftik, Yaniv Ben-Itzhak, and others.

This work is partly funded by EU SMARTEDGE project (*101092908*) & *VMWare. We acknowledge support from Intel, NVIDIA and AMD.*

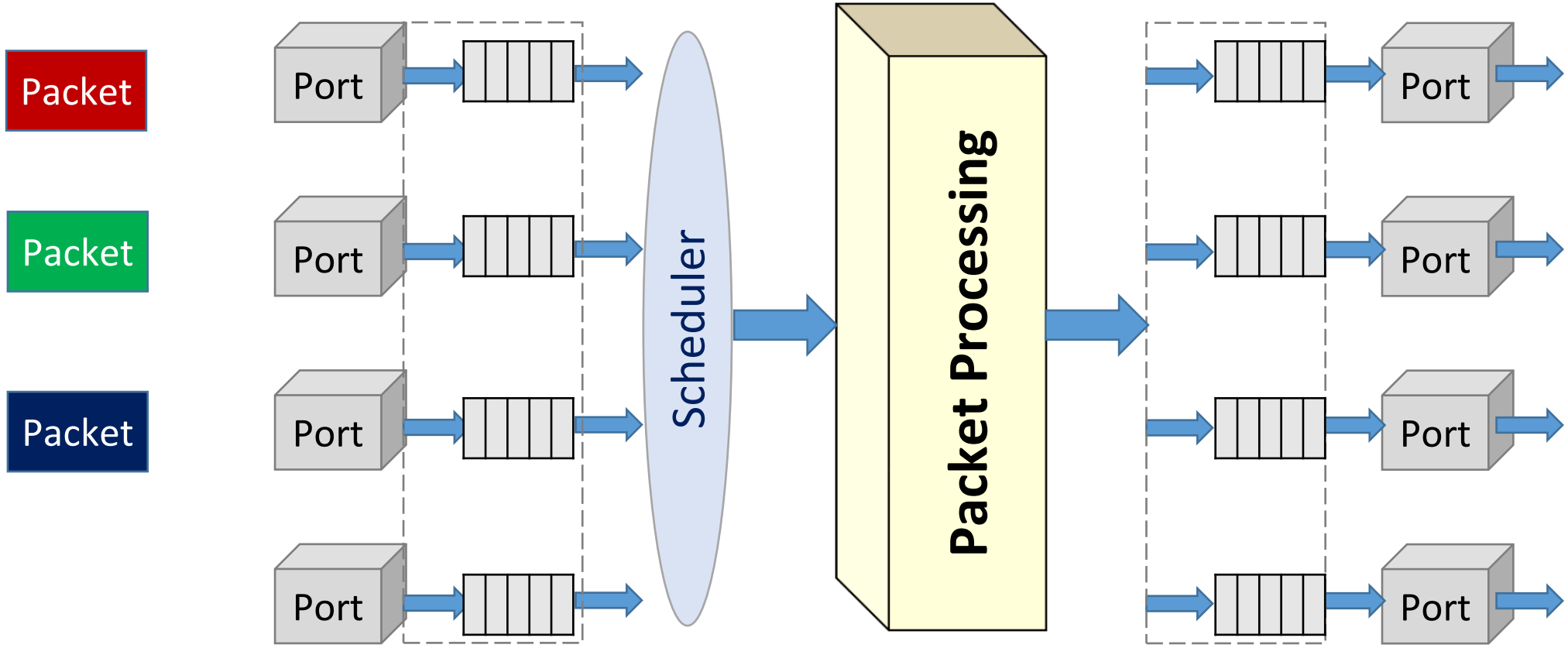
What is In-Network ML?

A brief introduction to network devices

(really brief)

Simplified Switch Architecture

To achieve high throughput, packet switches are pipelined



How fast is a switch?

- A single **device:**
  - 51.2Tbps - 64 x 800GE
  - > 10 billions packets per second
    - >1 TOPS

Simplified Programmable Packet Processing

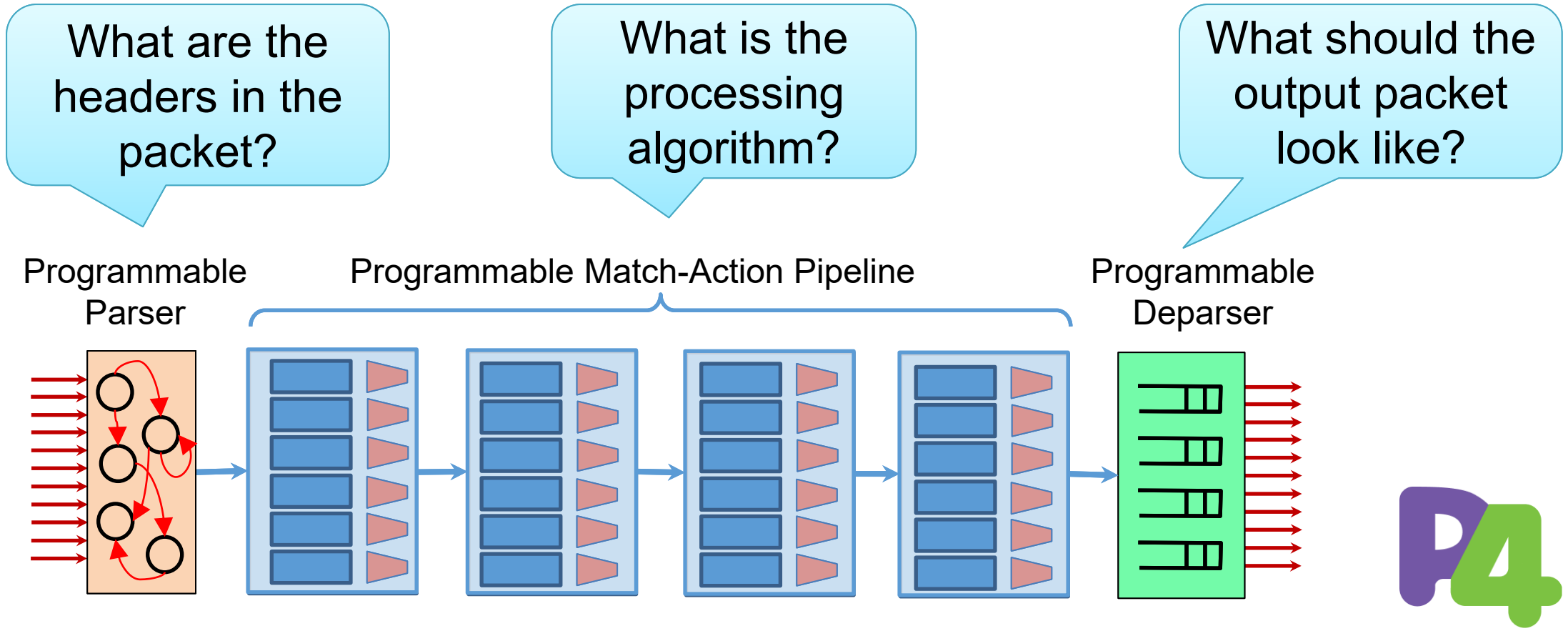


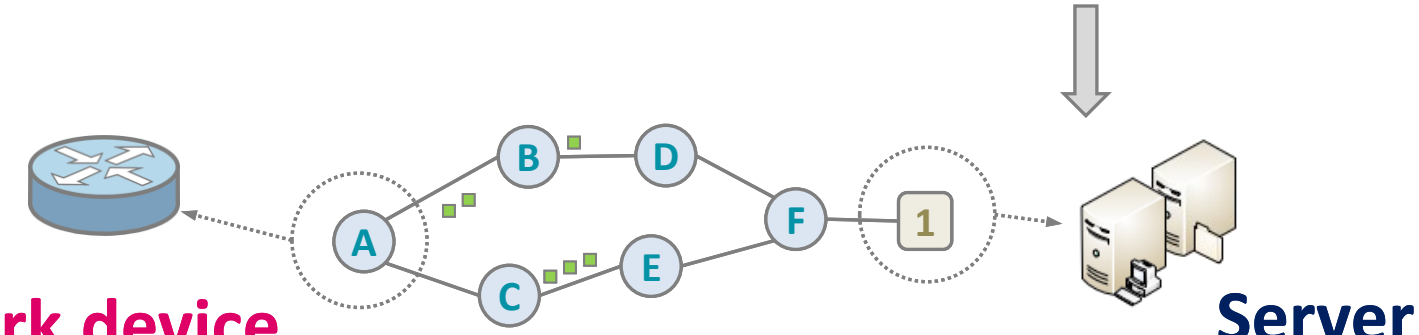
Figure source: p4.org

In-Network Machine Learning

Offload inference or entire ML processes to the network.

In-Network

Machine Learning Inference

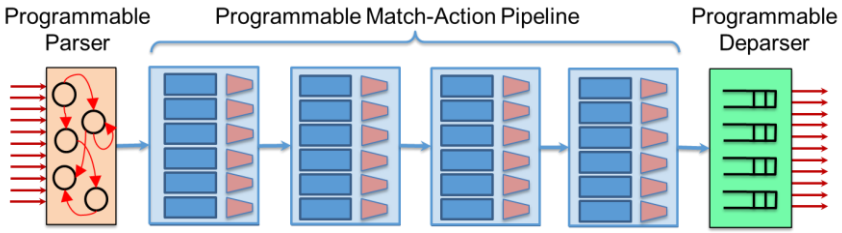


Network device

Server

Resource constrained ML

A network device is not a CPU / GPU!



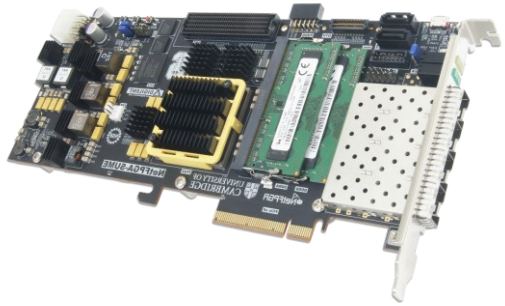
C. Zheng et al “In-Network Machine Learning Using Programmable Network Devices: A Survey,” IEEE Communications Surveys & Tutorials 2023.

Target Platforms

Switches

DPU/SmartNICs

FPGAs

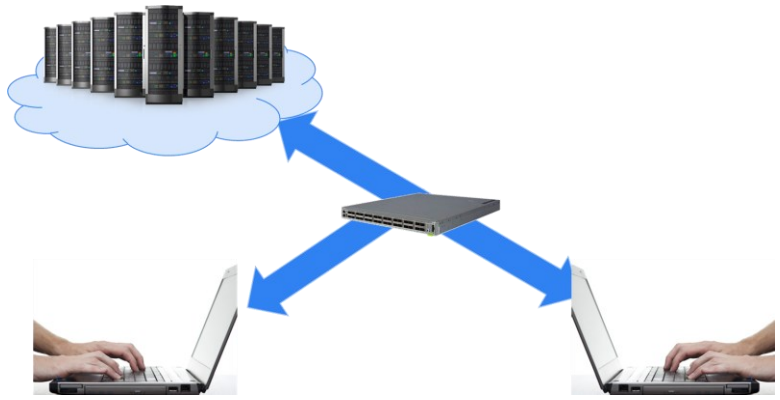


Low-end devices

C. Zheng et al “In-Network Machine Learning Using Programmable Network Devices: A Survey,” IEEE Communications Surveys & Tutorials 2023.

Motivation: The 3 Ls

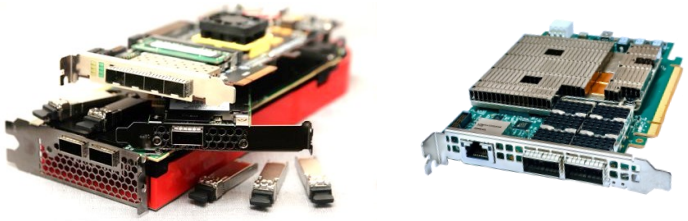
- **Location**
  - Along the path
  - Data aggregation
  - Already exists
- **Latency**
  - Early termination
- **Load**
  - Reduces load on servers / GPUs
  - High throughput



C. Zheng et al. “IIsy: Hybrid In-Network Classification Using Programmable Switches,” IEEE Transactions on Networking, 2024.

In-Network ML: Our Goals

- Run on commodity network devices
  - Off the shelf and unmodified!
- Co-exist with networking functionality
- Must not affect performance
- Code once, deploy across different devices
- Modularity
- Easy to Use



- **Stateless, no multiplication or loops, limited memory, different architecture ...**

In-Network ML: Goals and Non-Goals

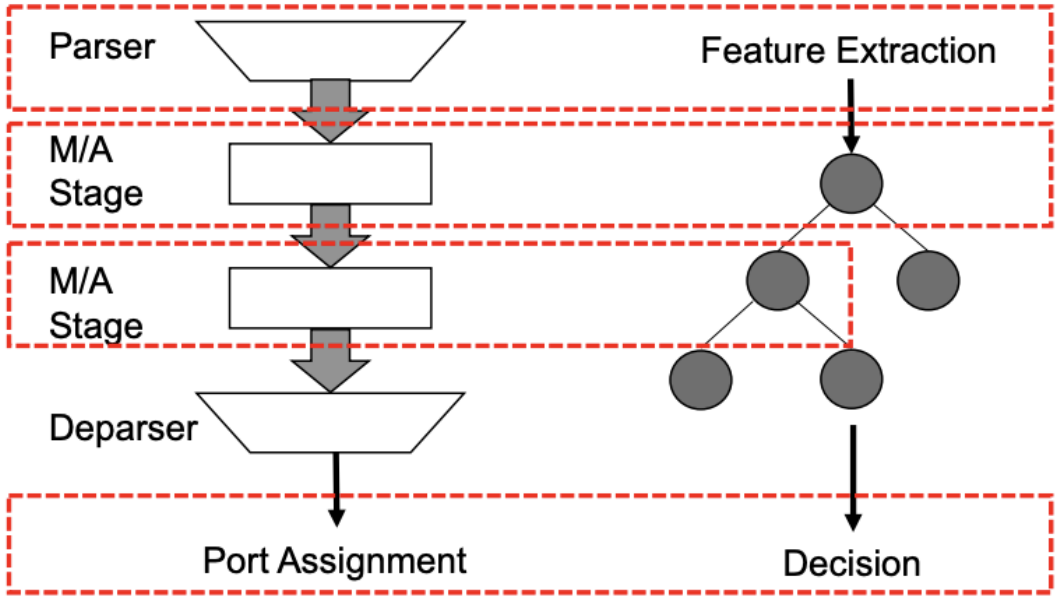
- Enable the technology
- Machine learning models:
  - Enable ***N*** different types of ML models
  - … but not necessarily latest or state of the art
- Machine learning performance (e.g., F1 score):
  - Similar to an **identical** model running on a server / GPU
  - … but less than a larger model running on a server / GPU
- Fit for purpose
  - ML Performance should be ***good enough*** for the use case
  - Provide a solution for improving ML performance
  - No compromise on ***system performance***

Three Mapping Methodologies

1. **Direct Mapping**
   - A series of sequential operations
   - Decision Tree, BNN, …
2. **Encode Based**
   - Slicing the feature space
   - K-means, Random Forest, …
3. **Look Up Based**
   - Use tables for math operations
   - Support Vector Machine, Naïve Bayes, …

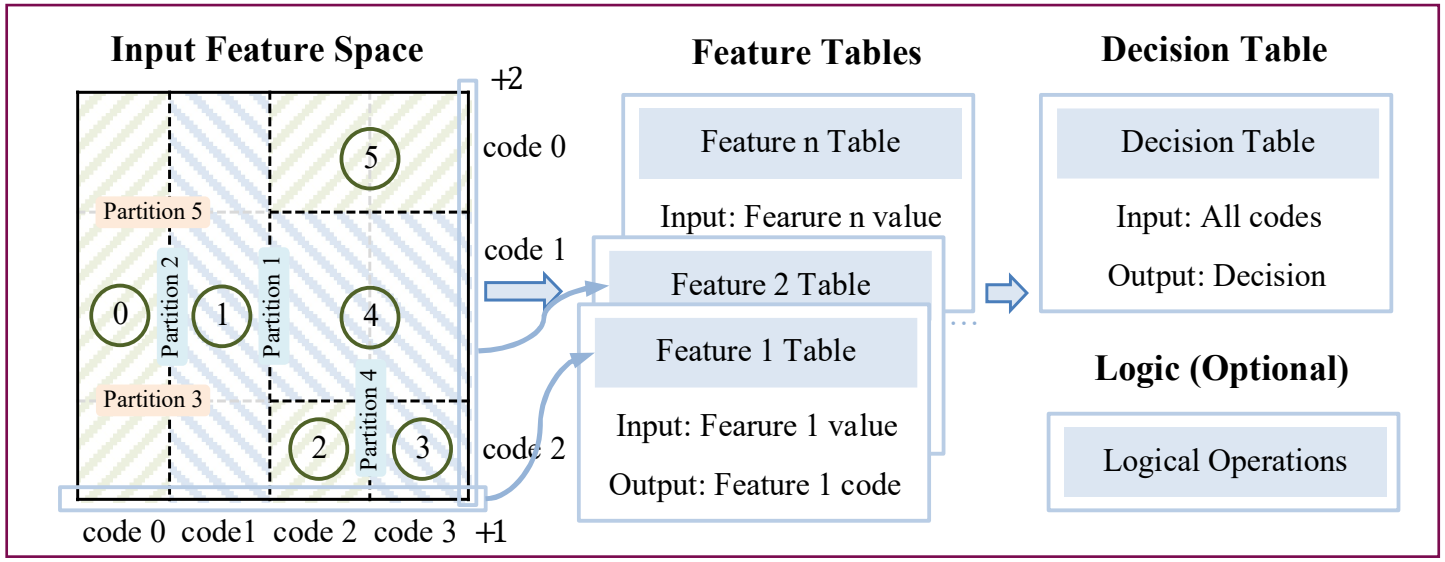
Three Mapping Methodologies

1. **Direct Mapping**
   - A series of sequential operations
   - Decision Tree, BNN, …
2. **Encode Based**
   - Slicing the feature space
   - K-means, Random Forest, …
3. **Look Up Based**
   - Use tables for math operations
   - Support Vector Machine, Naïve Bayes, …



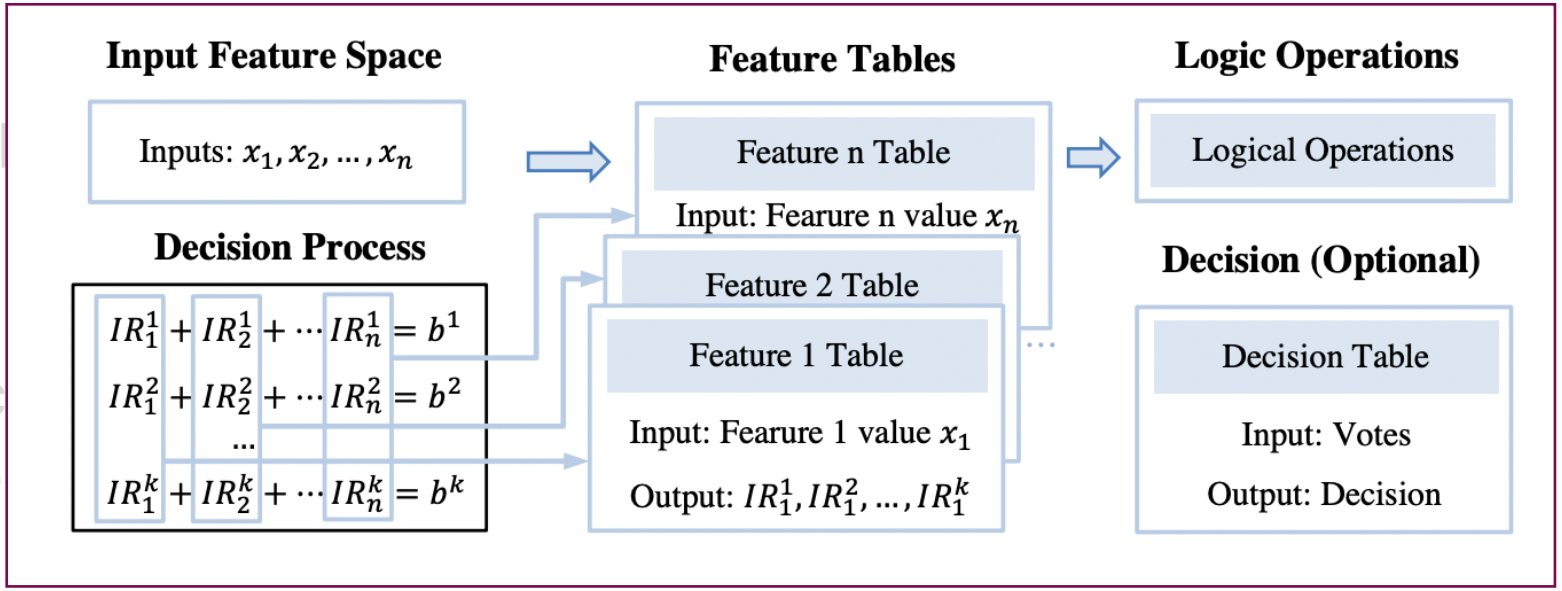
Three Mapping Methodologies

1. **Direct Mapping**
   - A series of sequential operat
   - Decision Tree, BNN, …
2. **Encode Based**
   - Slicing the feature space
   - K-means, Random Forest, …
3. **Look Up Based**
   - Use tables for math operations
   - Support Vector Machine, Naïve Bayes, …



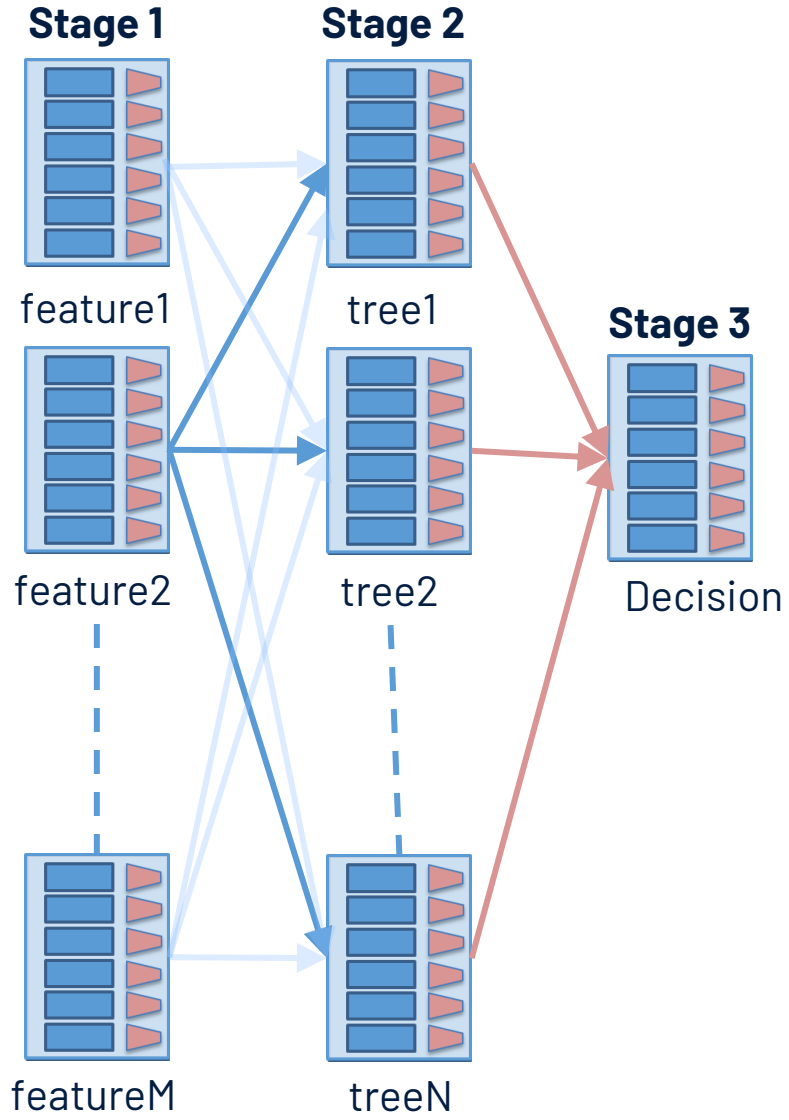
Three Mapping Methodologies

1. **Direct Mapping**
   - A series of sequential o
   - Decision Tree, BNN, …
2. **Encode Based**
   - Slicing the feature spac
   - K-means, Random Fore
3. **Look Up Based**
   - Use tables for math operations
   - Support Vector Machine, Naïve Bayes, …

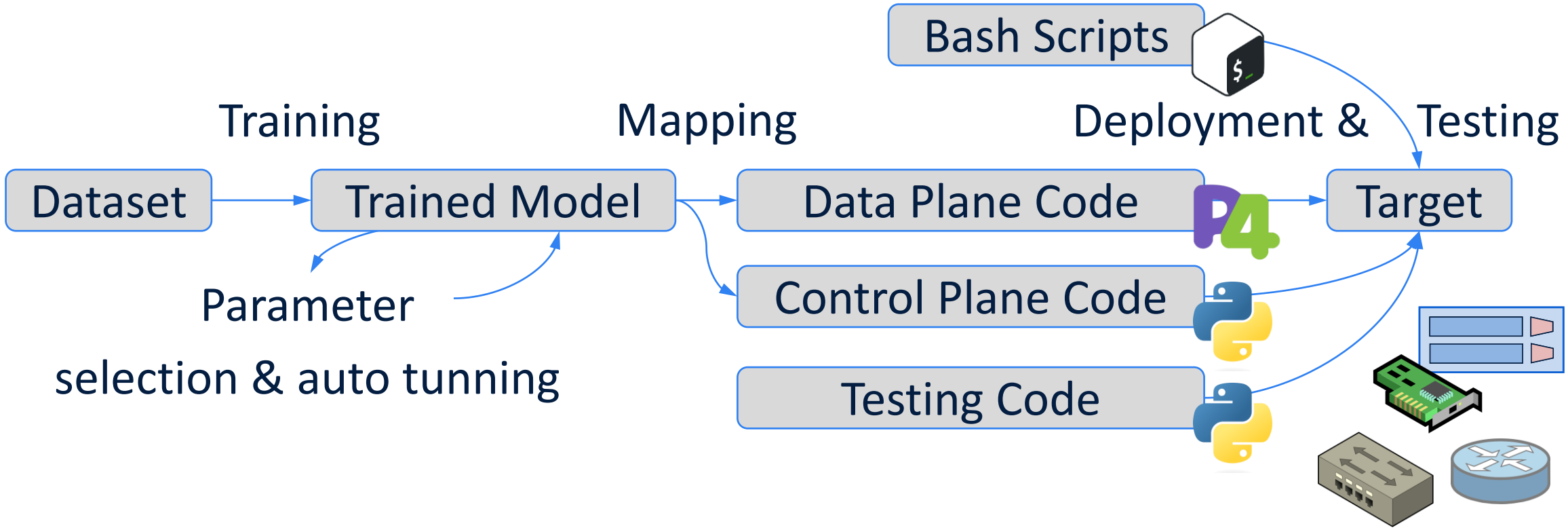


Mapping vs Resources

- Experience (switch-ASIC):
  - Stages and logic-per-stage are limiting
  - Memory is not as limiting
- Key: maximize independence, look up in parallel:
  - Features
  - Trees / hyperplanes / probabilities



Planter: Rapid Prototyping of In-Network ML



PLANTER

C. Zheng et al "Planter: rapid prototyping of in-network machine learning inference" CCR 2024.
https://github.com/In-Network-Machine-Learning/Planter

Planter: A Modular Framework

Models: SVM, Tree ensembles (Random Forest, XGBoost, …), K-Means, Naïve Bayes, KNN, PCA, Auto-Encoder, Neural Network, Q-Learning, …

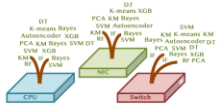
Targets: Switches (Intel), FPGA (AMD), DPU (NVIDIA), IoT Gateway (DELL), low cost (RaPi), software switch, …

ML Libraries: Scikit-learn, TensorFlow, …

Features: Packet-level, Flow-level, File (csv, json)

Datasets: UNSW, CICIDS, AWID3, KDD, NASDAQ, Requet, EDGEIIOT, Iris, …

Use Cases: Cybersecurity, Finance, IoT, Smart Grid, Manufacturing, Networking, …



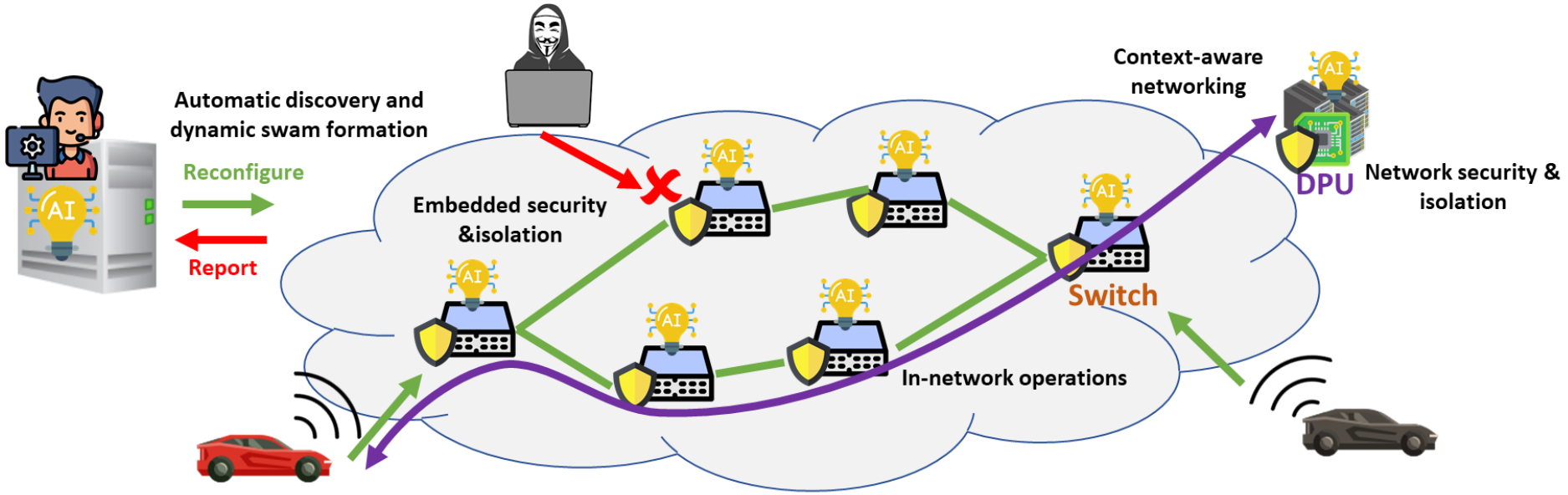
PLANTER

C. Zheng et al "Planter: rapid prototyping of in-network machine learning inference" CCR 2024.
https://github.com/In-Network-Machine-Learning/Planter

Anomaly Detection in SmartEdge

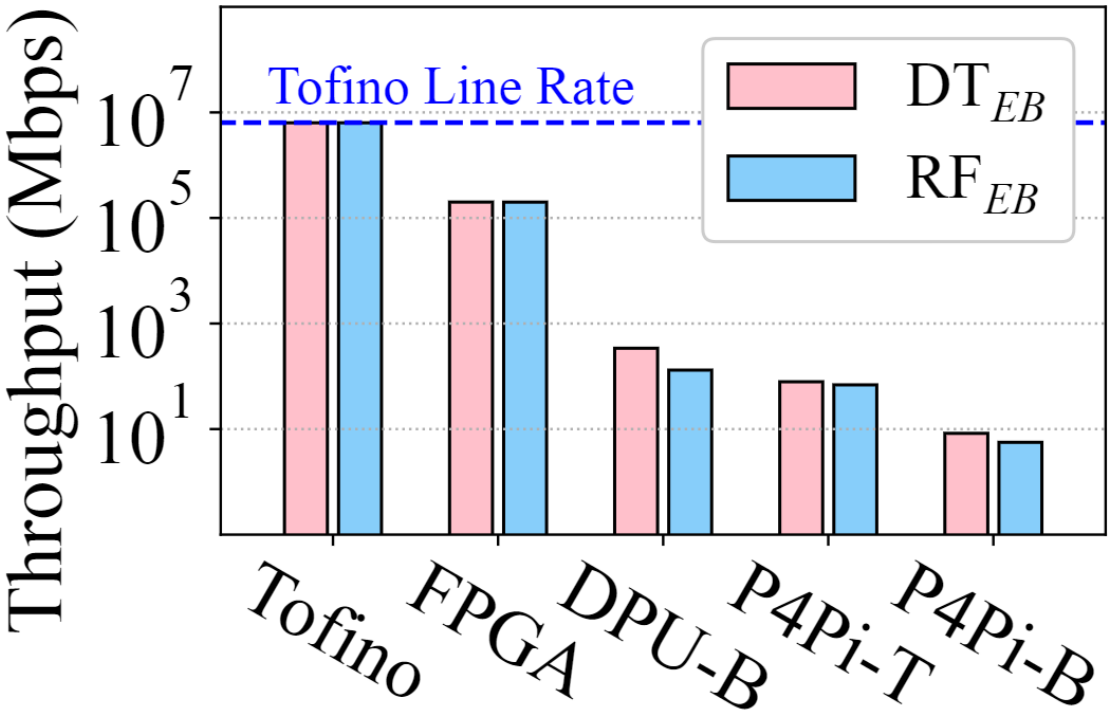
Improving dynamic swarms' operation

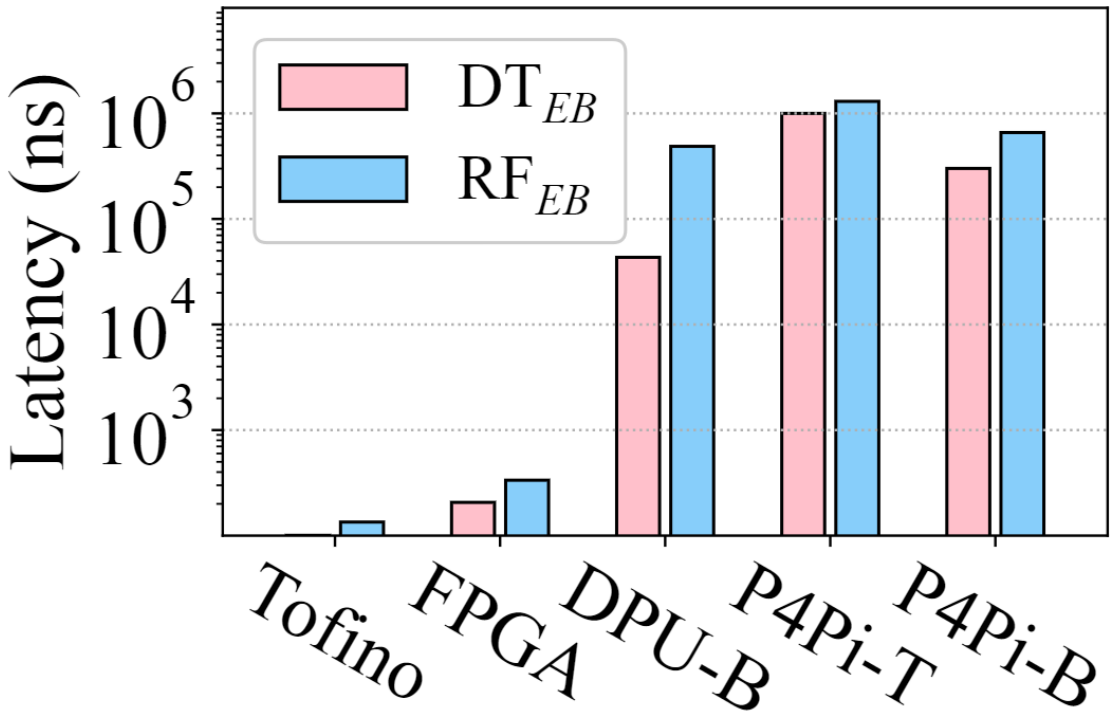
Using in-network ML to react *instantly* to incidents, security threats, or changes in operating conditions



PLANTER

System Performance



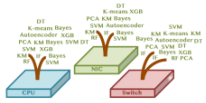


PLANTER

C. Zheng et al "Planter: rapid prototyping of in-network machine learning inference" CCR 2024.
https://github.com/In-Network-Machine-Learning/Planter

System Performance – Switch ASIC vs FPGA

	FPGA (Alveo U280)	Switch (Intel Tofino)
Throughput	100Gb/s	64x 100Gb/s
Added latency	170ns-320ns	~0ns- <1µs
Memory	Up to GBs	Up to 10's of MBs
Externs	(semi) programmable	Fixed
Utilization (typical)	6%-7% LUT ~4% RAM	0-4 stages, 1%-5% Memory



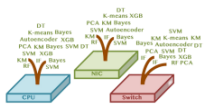
PLANTER

C. Zheng et al "Planter: rapid prototyping of in-network machine learning inference" CCR 2024.
https://github.com/In-Network-Machine-Learning/Planter

ML Performance

Anomaly Detection, Random Forest, confidence threshold 0.7

	Small	Medium	Large	Baseline
Features	4	5	6	25
Trees	6	10	14	200
Max Depth	4	5	6	—
Accuracy	97.05	97.17	97.78	99.51
Precision	98.06	98.12	98.60	99.67
Recall	88.55	89.04	91.36	99.75
F1 score	92.60	92.94	94.58	98.88

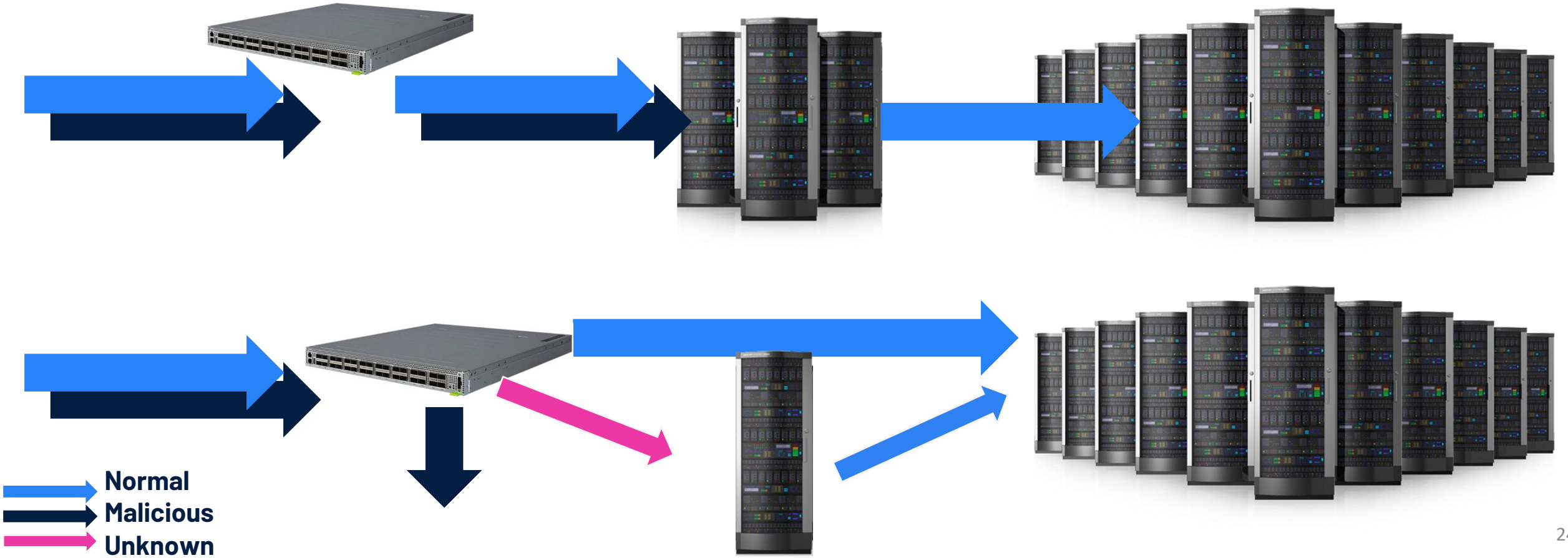


PLANTER

Zheng et al, IIsy: Hybrid In-Network Classification Using Programmable Switches, 2024

Anomaly Detection – Hybrid Deployment

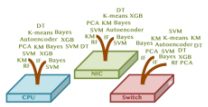
Goal: increasing ML performance & reducing back-end resources



ML Performance

Anomaly Detection, Random Forest, confidence threshold 0.7

	Small	Medium	Large	Baseline
Features	4	5	6	25
Trees	6	10	14	200
Max Depth	4	5	6	—
Accuracy	97.05	97.17	97.78	99.51
Precision	98.06	98.12	98.60	99.67
Recall	88.55	89.04	91.36	99.75
F1 score	92.60	92.94	94.58	98.88
Hybrid Accuracy	98.58	98.94	99.31	—
Hybrid F1	96.64	97.53	98.41	—

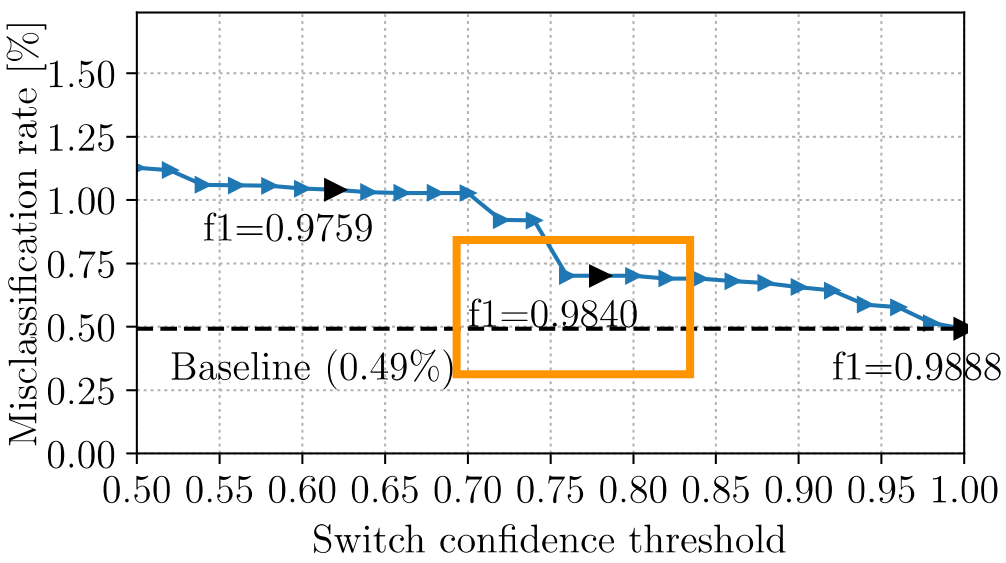


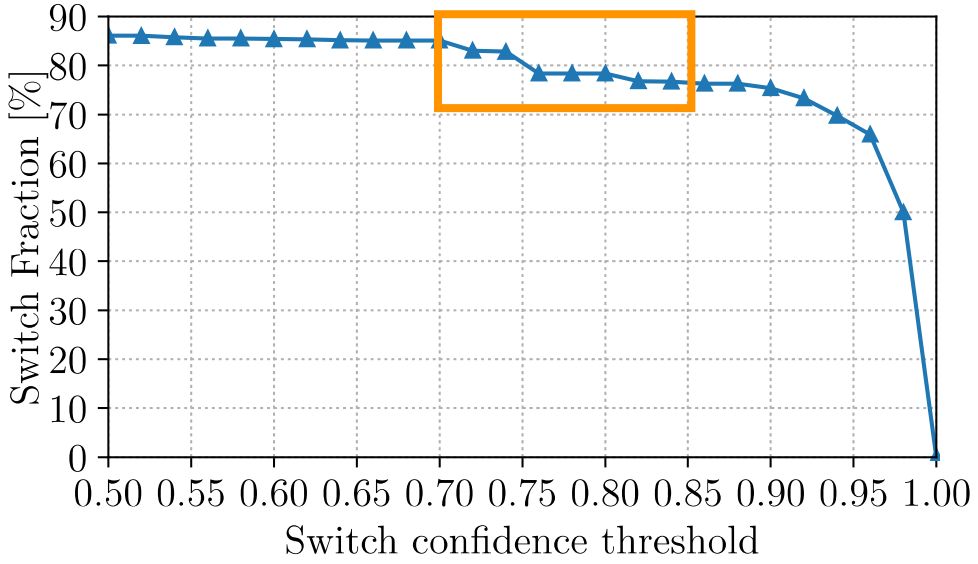
PLANTER

Zheng et al, IIsy: Hybrid In-Network Classification Using Programmable Switches, 2024

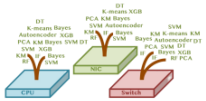
Anomaly Detection - Hybrid Model

Same model in a hybrid deployment





Error Rate & Fraction of Traffic Handled by the Switch vs Switch Confidence Threshold



PLANTER

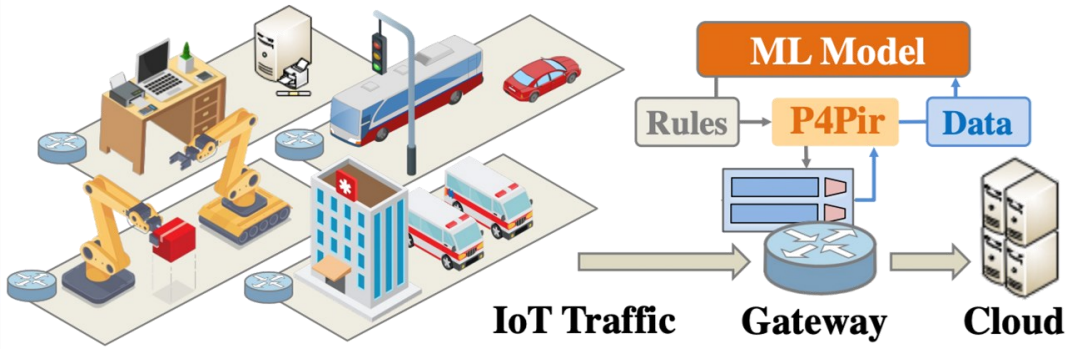
Zheng et al, IIsy: Hybrid In-Network Classification Using Programmable Switches, 2024

Example: Traffic Analysis for Smart IoT Gateways

Terminate data at the IoT Gateway

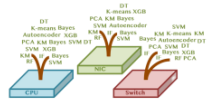
- SmartEdge smart factories use case
- Operate on IoT and sensor data
- Provide continuous threat defence
  - In-band feature extraction and mitigation
  - Proactive logging
  - Unsupervised labeling of traffic
  - Continuous updates of in-network model
- Federated learning using multiple gateways
- Runs on P4Pi (P4 on Raspberry Pi) and DELL IoT Gateway 5200

IoT Gateway



Zang et al, Towards Continuous Threat Defense: In-Network Traffic Analysis for IoT Gateways, 2023

Zang et al, Federated In-Network Machine Learning for Privacy-Preserving IoT Traffic Analysis, 2024

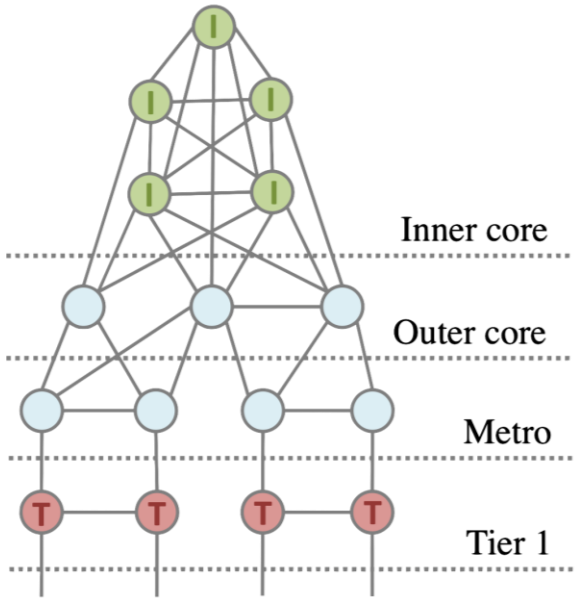


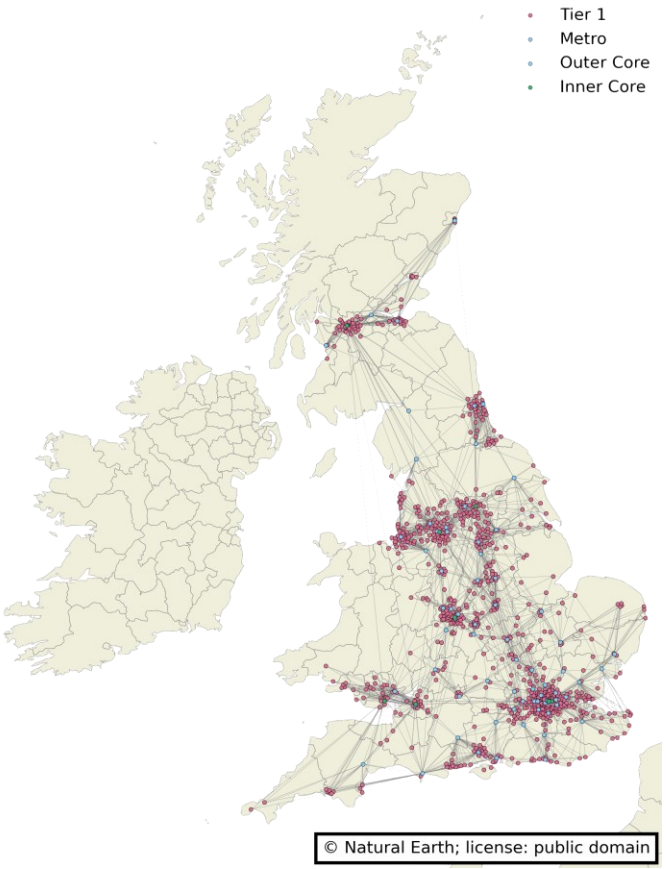
PLANTER

Example: Attack Detection on BT Network

Distributed ML Deployment

- Any path through the network
- Without affecting existing network functions.
- Information sharing across nodes



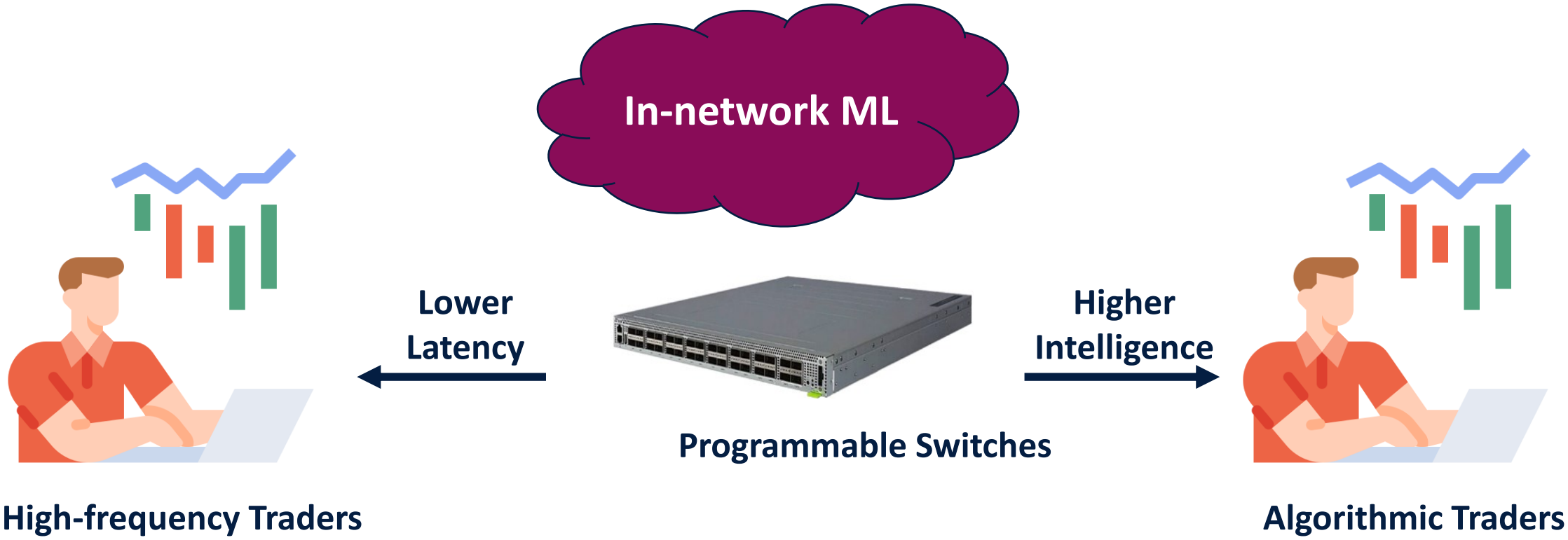


BT Network Topology
Backbone WAN with 1008 Nodes

PLANTER

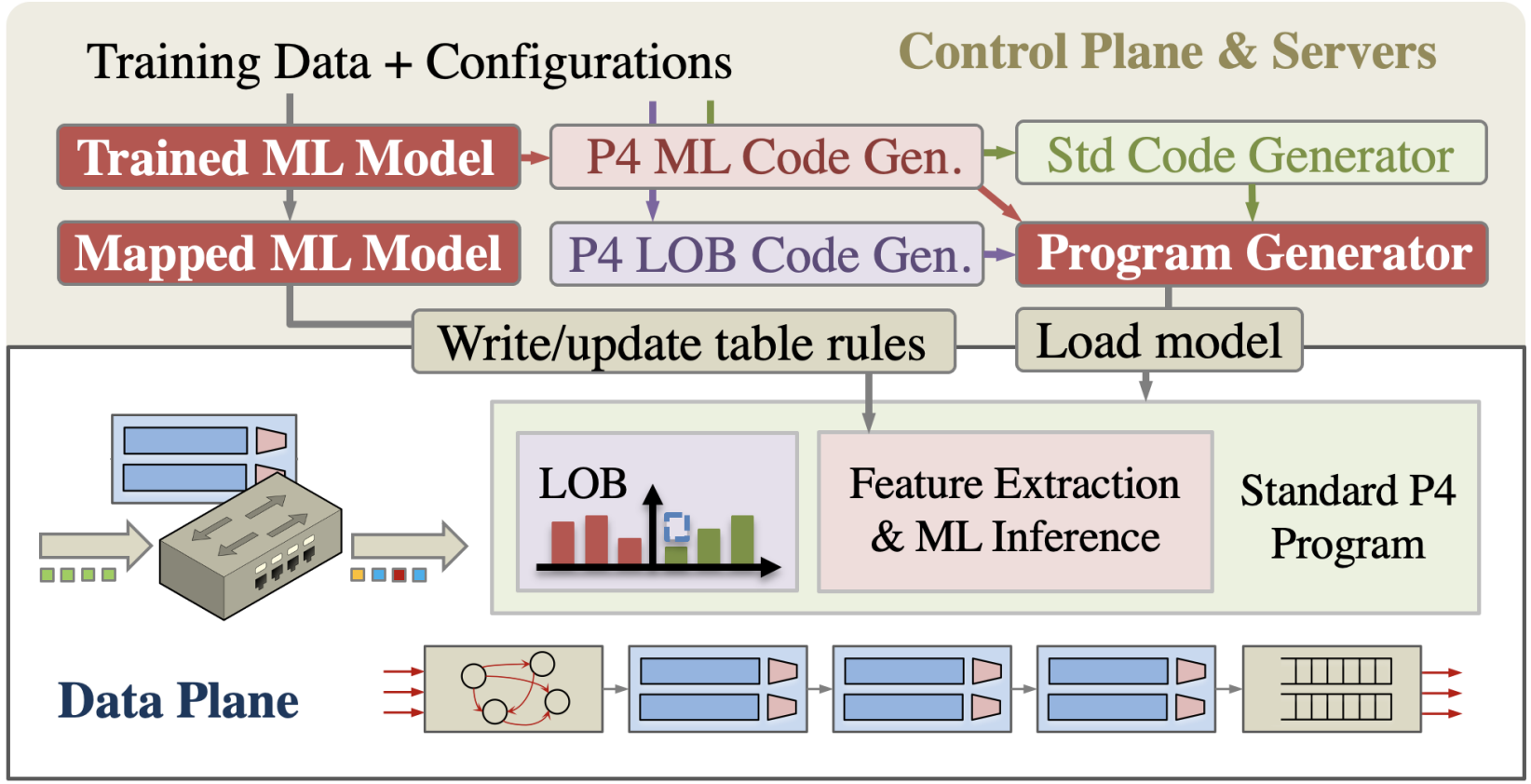
Zheng et al, DINC: Toward Distributed In-Network Computing, 2023

Example: Price Movement Forecasting



Hong et al., In-Network Machine Learning for Limit Order Books, HPSR 2023

LOBIN: In-Network Price Movement Forecasting



45% of traffic and 1.97 Billion USD per day processed on a switch.

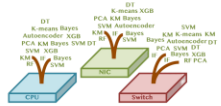
Hong et al., In-Network Machine Learning for Limit Order Books, HPSR 2023

Summary

Moving Intelligence to the Network

- Commodity network devices as inference engines
- Support of:
  - Rapid prototyping on a range of network devices
  - High throughput, low latency
  - Modular: “bring your own model”
  - Distributed, federated and hybrid deployments
- A lot left to explore, try and research! Code is open!

https://github.com/In-Network-Machine-Learning/Planter



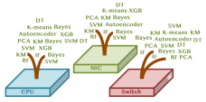
PLANTER

More Information

https://eng.ox.ac.uk/computing/projects/in-network-ml/
planter@eng.ox.ac.uk

- **Publications (Selected):**
  - C. Zheng et al, "Planter: rapid prototyping of in-network machine learning inference", Computer Communication Reviews, 2024.
  - C. Zheng et al, “Ilsy: Hybrid In-Network Classification Using Programmable Switches,” IEEE Transactions on Networking, 2024.
  - C. Zheng et al, “In-Network Machine Learning Using Programmable Network Devices: A Survey,” IEEE Communications Surveys & Tutorials, 2023.
  - M. Zang et al, “Towards Continuous Threat Defense: In-Network Traffic Analysis for IoT Gateways,” IEEE IoT Journal, 2023.
  - C. Zheng et al, DINC: Toward Distributed In-Network Computing, ACM CoNEXT, 2023.
  - X. Hong et al, "In-Network Machine Learning for Real-Time Transaction Fraud Detection“, ECAI 2024.
  - M. Zang et al, "Federated In-Network Machine Learning for Privacy-Preserving IoT Traffic Analysis", ACM TIOT, 2024.
  - M. Hemmatpour et al, "GridWatch: A Smart Network for Smart Grid“, IEEE SmartGridComm 2024.
- **Open source repositories:**
  - Planter: https://github.com/In-Network-Machine-Learning/Planter
  - Ilsy: https://github.com/In-Network-Machine-Learning/Ilsy
  - DINC: https://github.com/In-Network-Machine-Learning/DINC
  - P4Pir: https://github.com/In-Network-Machine-Learning/P4Pir
  - QCMP: https://github.com/In-Network-Machine-Learning/QCMP

We are recruiting!
https://eng.ox.ac.uk/computing/join-us/



PLANTER