Is In-network Machine Learning So Easy?

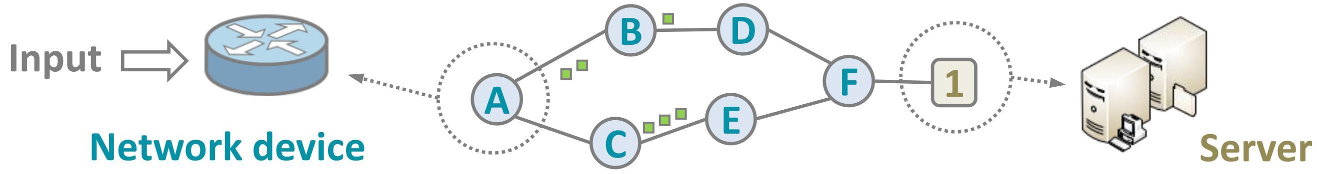
Changgang Zheng, University of Oxford

Joint work with Mingyuan Zang (DTU), Xinpeng Hong, Riyad Bensoussane, Liam Perreault (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)

Coseners 2023

Background

Growth of network connections & data



Growth of programmable data plane architecture

Traditional Network

- Bound to specific hardware
- Limited programmability
- High barrier to modification

Programmable Network

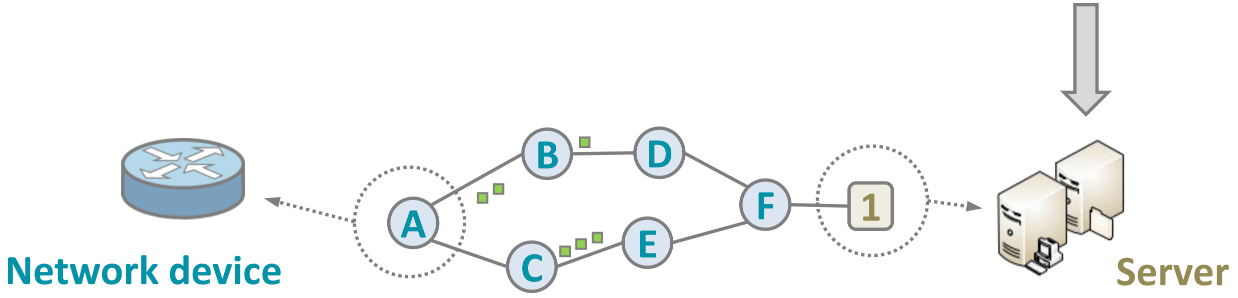
- Different hardware, same architecture
- Enable programmable control
- Easy and centralised control and modification

What Is In-Network Machine Learning?

In-network ML refers to off load inference or entire ML processes to the network.

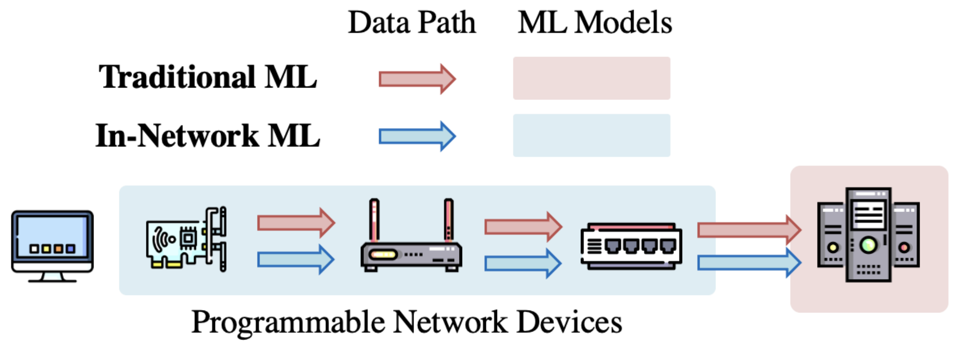
In-Network

Machine Learning Inference



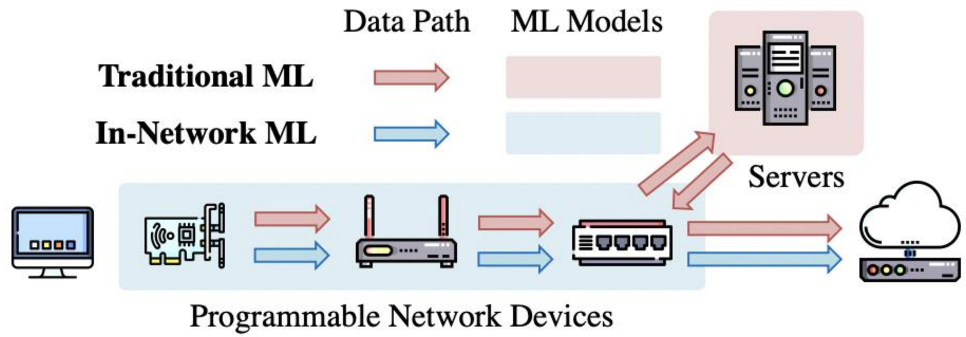
Motivation: 3Ls

Location



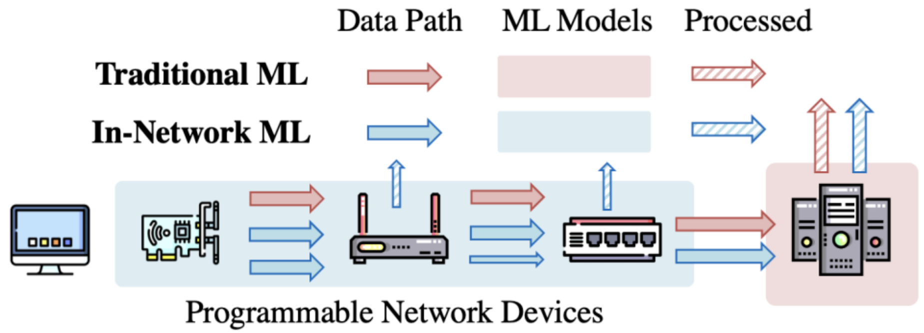
- Along the path
- Already exist

Latency



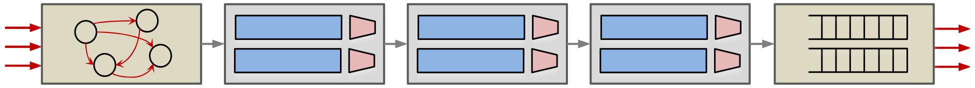
- Shorter path

Load



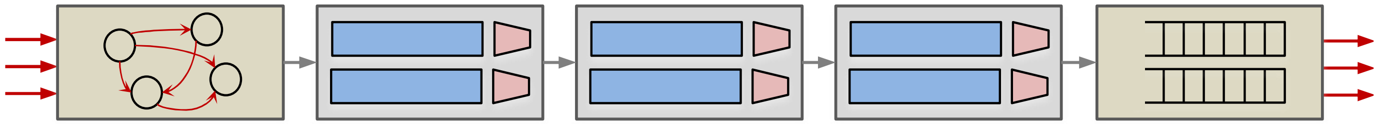
- Higher throughput
- Early termination

What Is In-Network Machine Learning?

General Machine Learning	**vs**	***In-Network Machine Learning***
Local PC, Servers, …	Location	*Network Infrastructures*
CPU, GPU, …	Device	 **PISA**
C, Python, MATLAB, …	Language	*P4*
Training & Inference	Manner	*Offline Training Online Inference*

Challenges

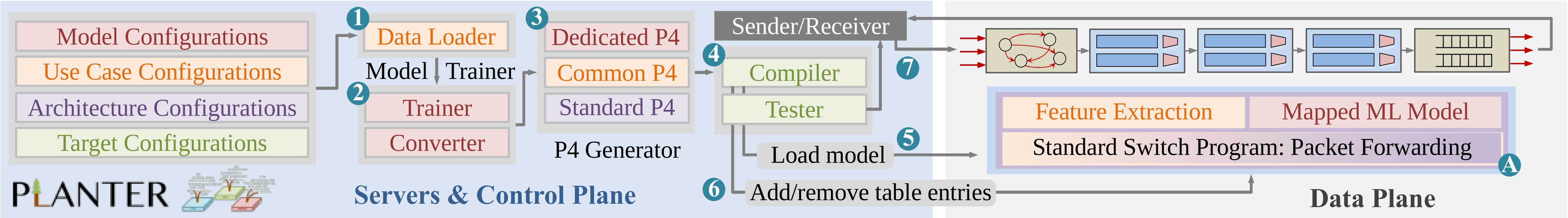
Resources on network devices are very limited compared to PC or servers.



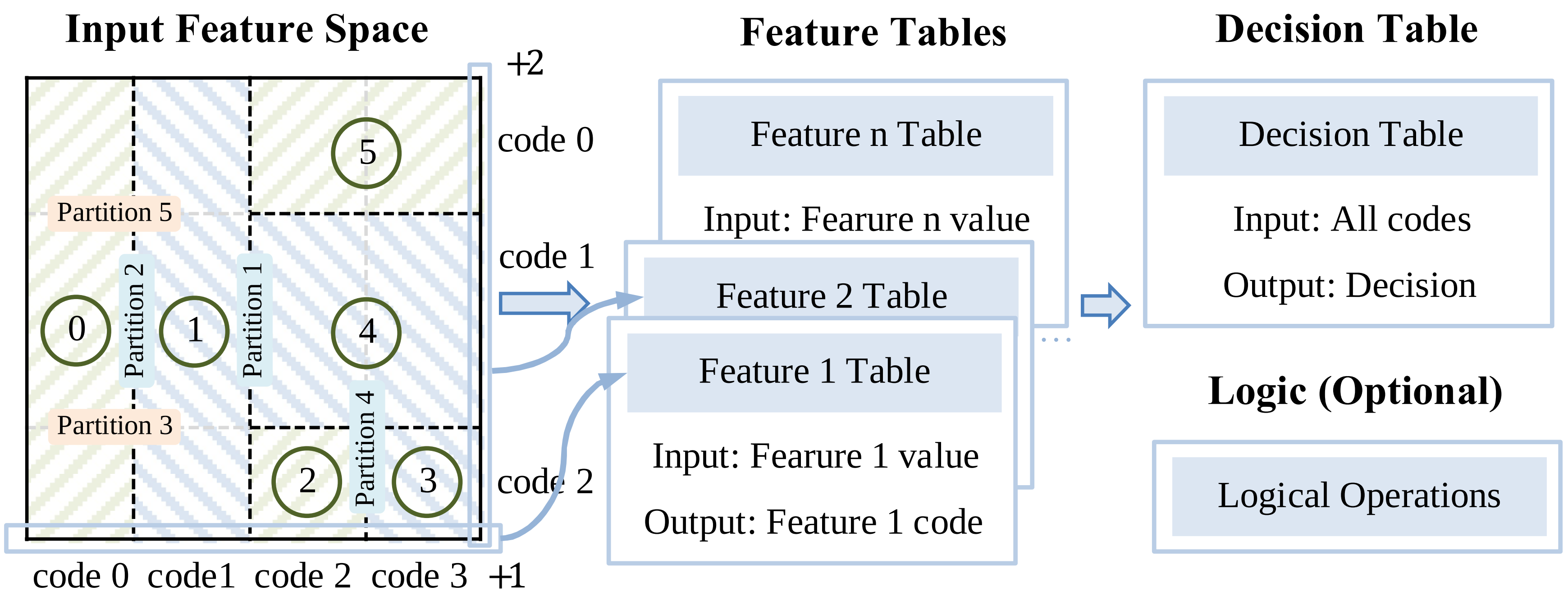
1. **Limited mathematical operations**
2. **Limited memory**
3. **Limited data types**
4. **Limited stages**

How to map?

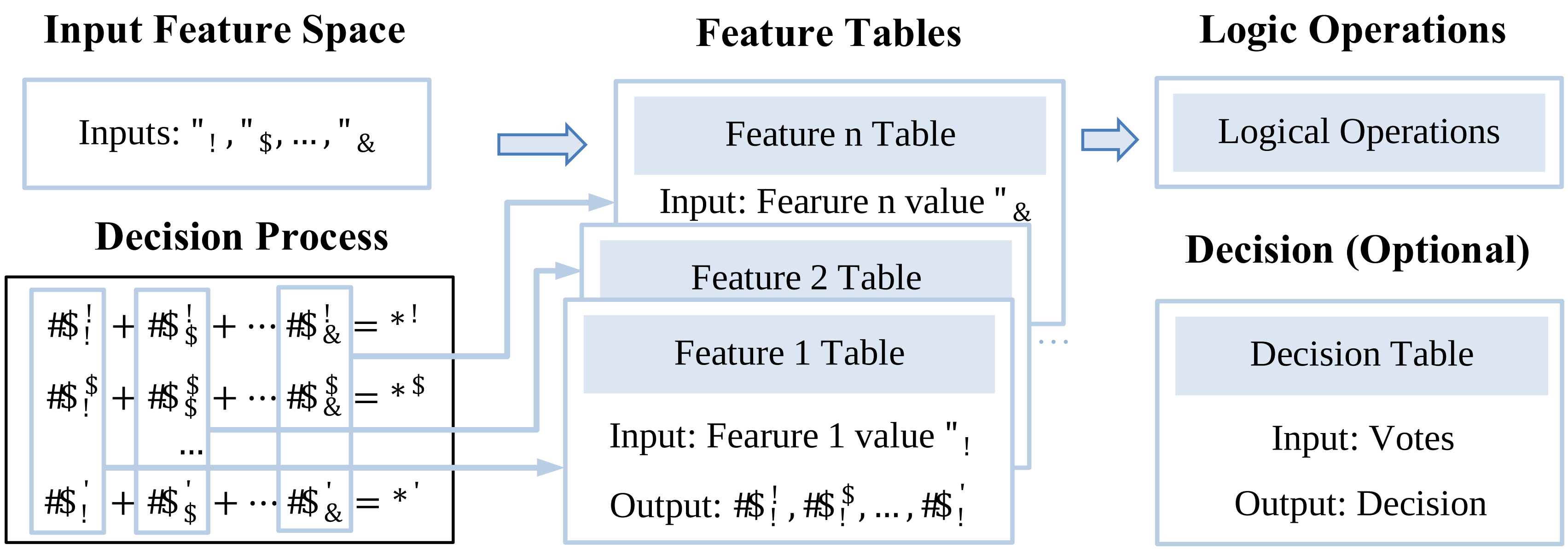
1. **Direct mapping solution**
2. **Encode based solution**
3. **Look up based solution**



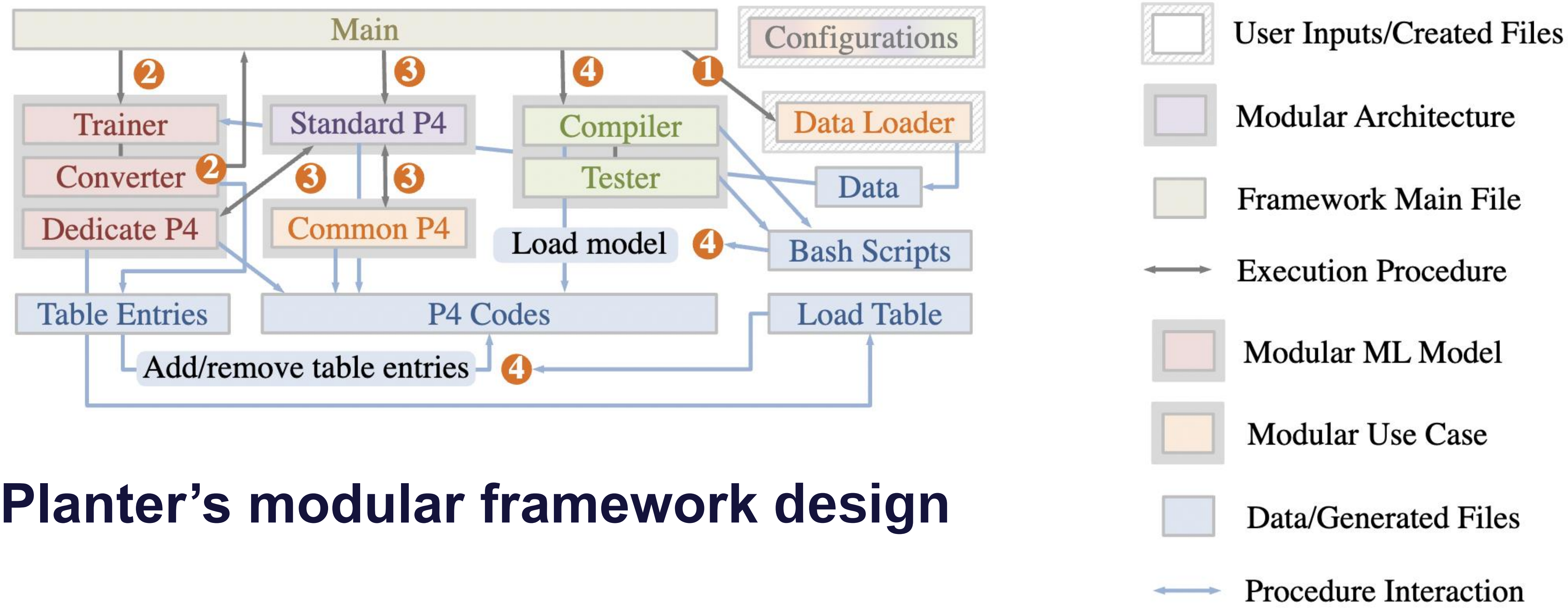
Encode based solution



Look-up based solution

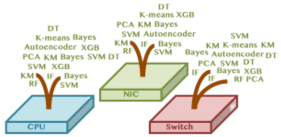


Planter Framework



Planter's modular framework design

Planter Framework



Models: SVM, DT, RF, XGB, IF, NB, KM, KNN, PCA, AE, NN

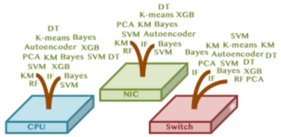
Architectures: PSA, v1model, TNA

Targets: Tofino, BMv2, P4Pi, Alveo FPGA

Datasets: Iris, UNSW, CICIDS, AWID3, KDD …

Use Cases: Anomaly Detection, Financial Transaction …

Planter Use Cases



PLANTER

DEPARTMENT OF ENGINEERING SCIENCE

UNIVERSITY OF OXFORD

Anomaly Detection

1. Models Mapping
2. Planter Framework
3. **Packet/Flow/File Level**

IIsy **Planter**

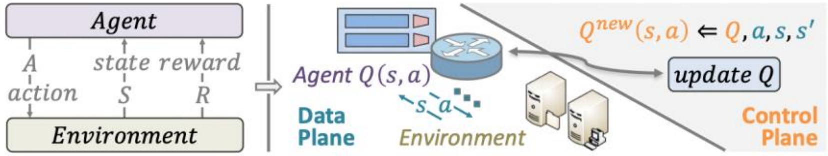
IoT Traffic Classification

1. **Continuous learning**
2. **Runtime model update**
3. **Federated learning**

P4Pir **FLIP4**

Load Balancing

1. **In-network Q-Learning**
2. **QCMP Load Balancing**

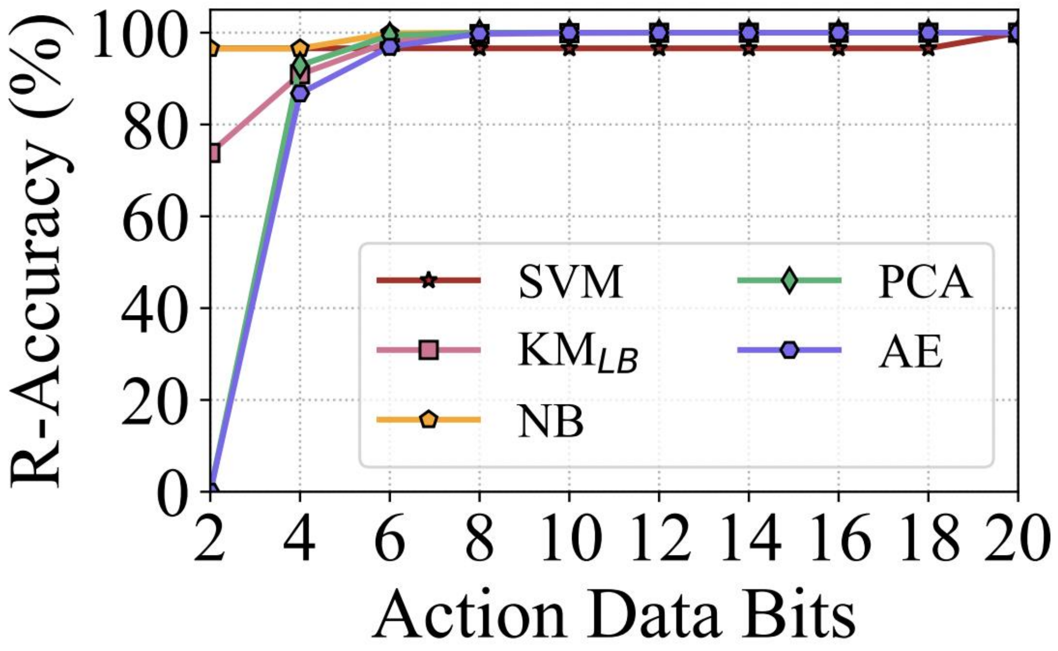


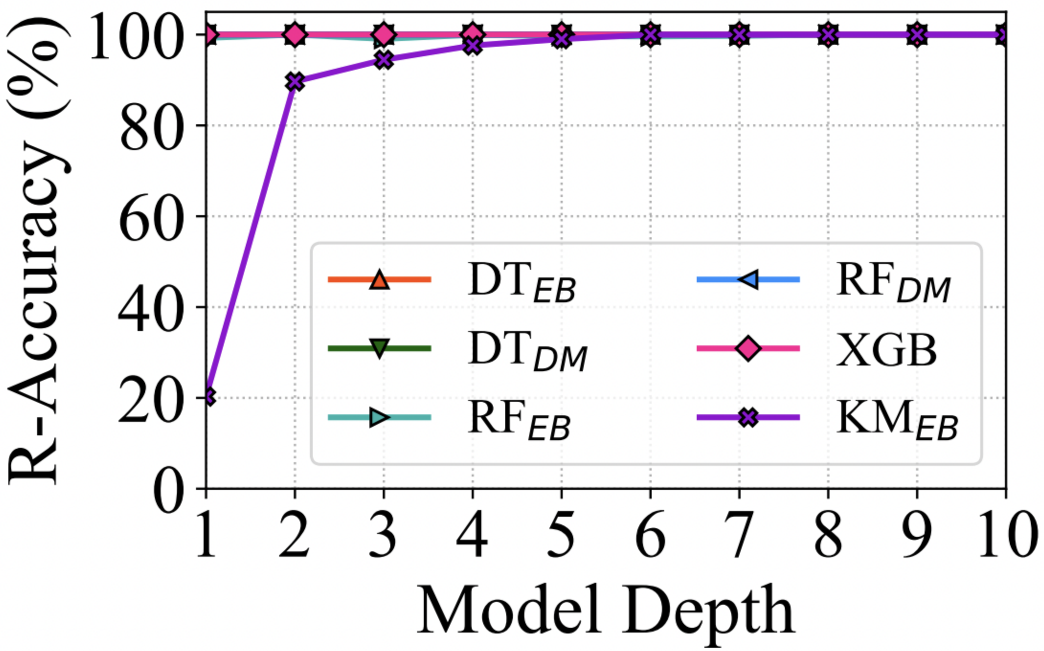
QCMP: SIGCOMM23 FIRA Workshop

Edge Computing, Financial Market Prediction…

Planter Results: Same Accuracy?

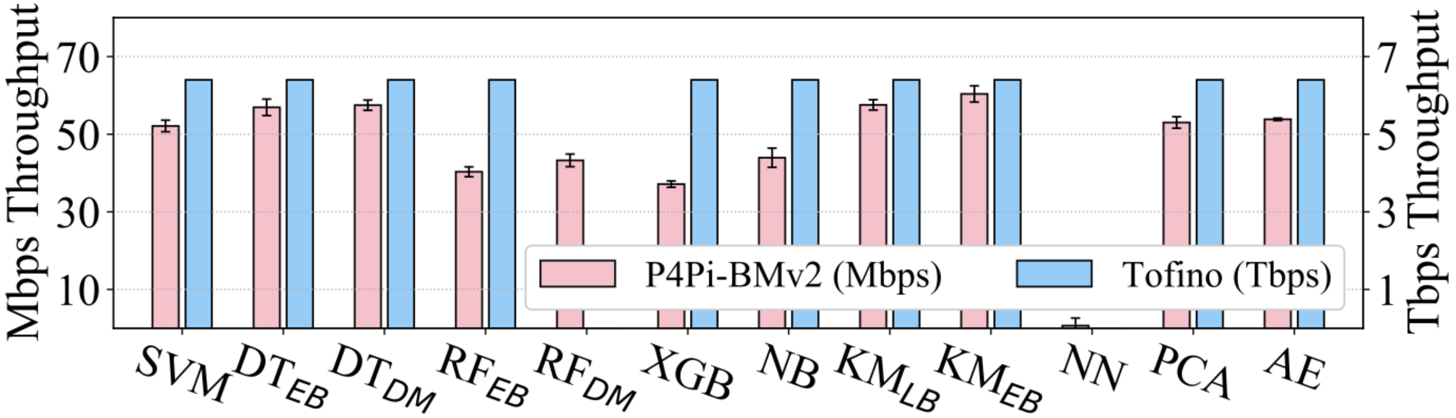
Anomaly Detection Use Case



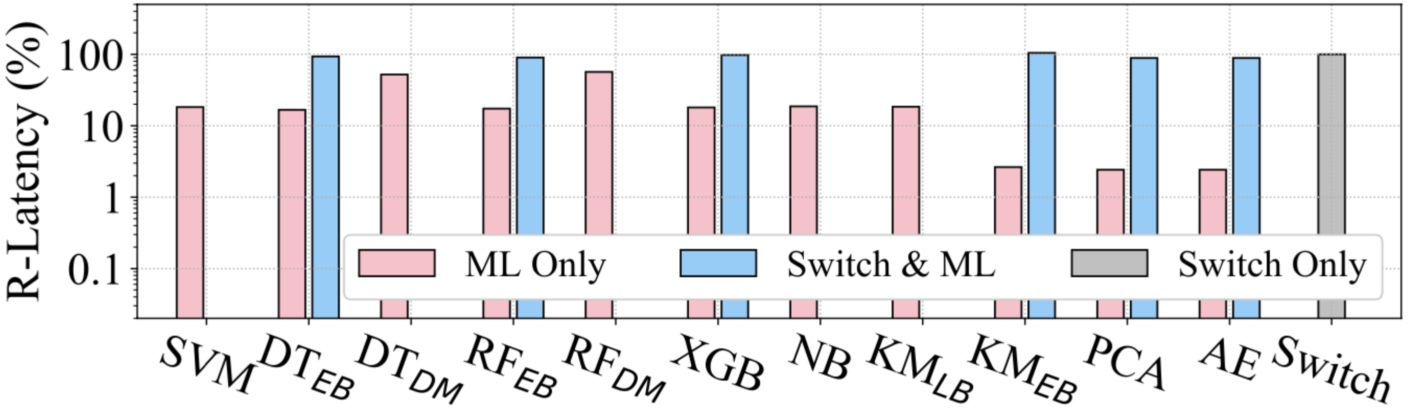


Planter Results: System Performance

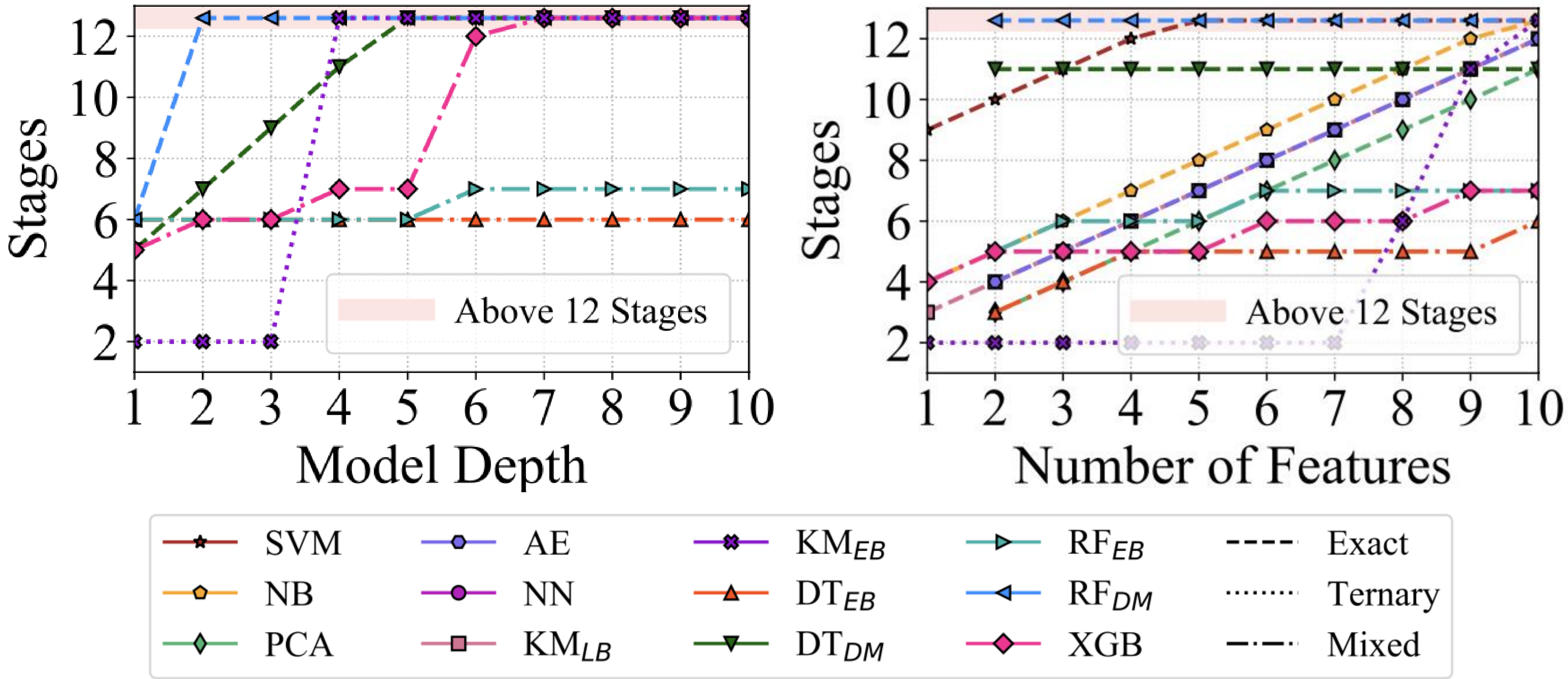
Anomaly Detection Use Case



Financial Market Prediction Use Case

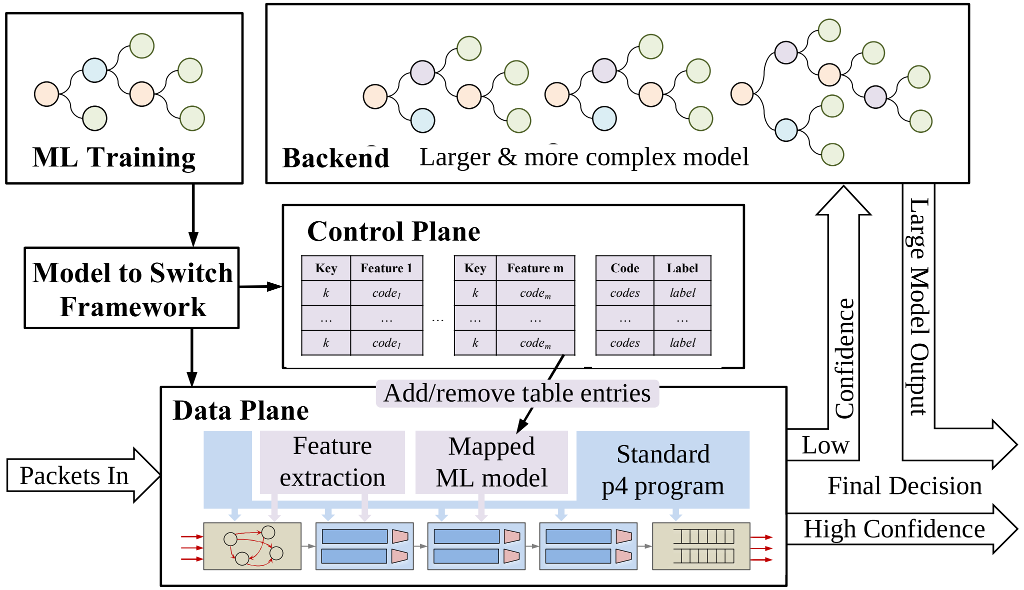


Planter Results: Scalability

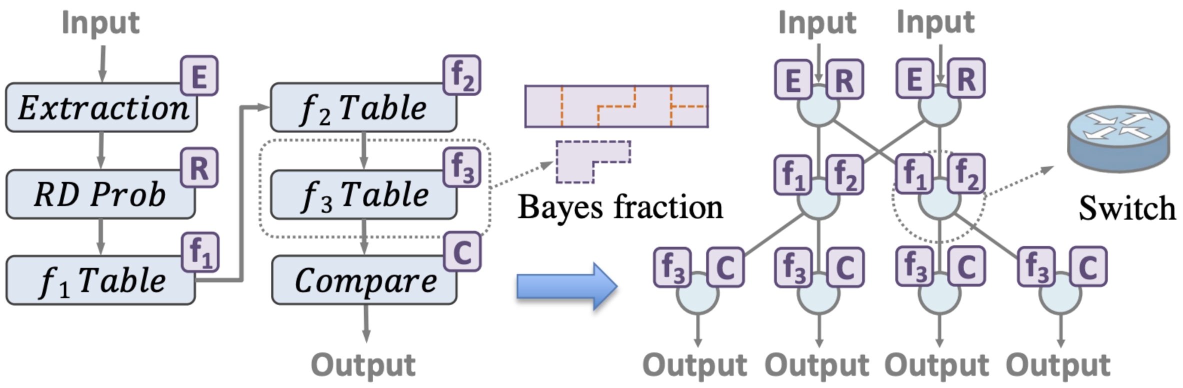


Further Scale In-Network ML?

Hybrid Deployment



Distributed Deployment



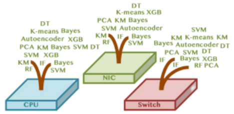
Summary

Q: How to realize in-network ML mapping?

A: Three mapping solution: DM, EB, LB.

Q: How to easily map ML to the data plane?

A: Planter framework.



PLANTER

Q: How to realize personalized use cases?

A: By adding new modules.

Q: How to further scale ML model size?

A: Hybrid deployment & distributed deployment.

Use Case Ideas? New Challenges?

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

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, IIsy: 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
Hong et al, LOBIN: In-Network Machine Learning for Limit Order Books, 2023
Zang et al, Federated Learning-Based In-Network Traffic Analysis on IoT Edge, 2023
Zheng et al, QCMP: Load Balancing via In-network Reinforcement Learning, 2023

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