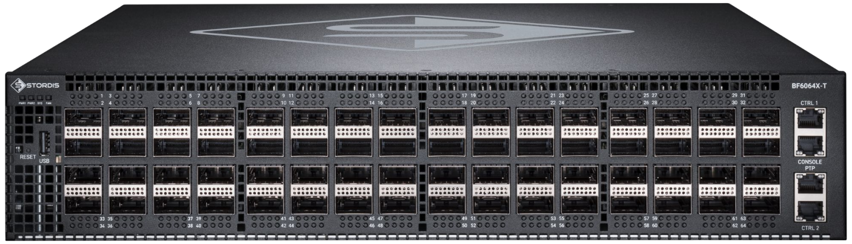
Henna: Hierarchical Machine Learning Inference in Programmable Switches

Aristide T-J. Akem[1,2], Beyza Bütün[1,2], Michele Gucciardo[1] and Marco Fiore[1]
[1]IMDEA Networks Institute, [2]Universidad Carlos III de Madrid

Native Network Intelligence Workshop, 9 December 2022

Developing the Science of Networks



Programmable
High Throughput
Low Latency

Applications

Sketches

Flow Monitoring

Traffic Management

Routing and Forwarding

Service Function Chaining

Time-Sensitive Networking

In-Band Network Telemetry

Network Function Virtualization

Intrusion Detection Systems

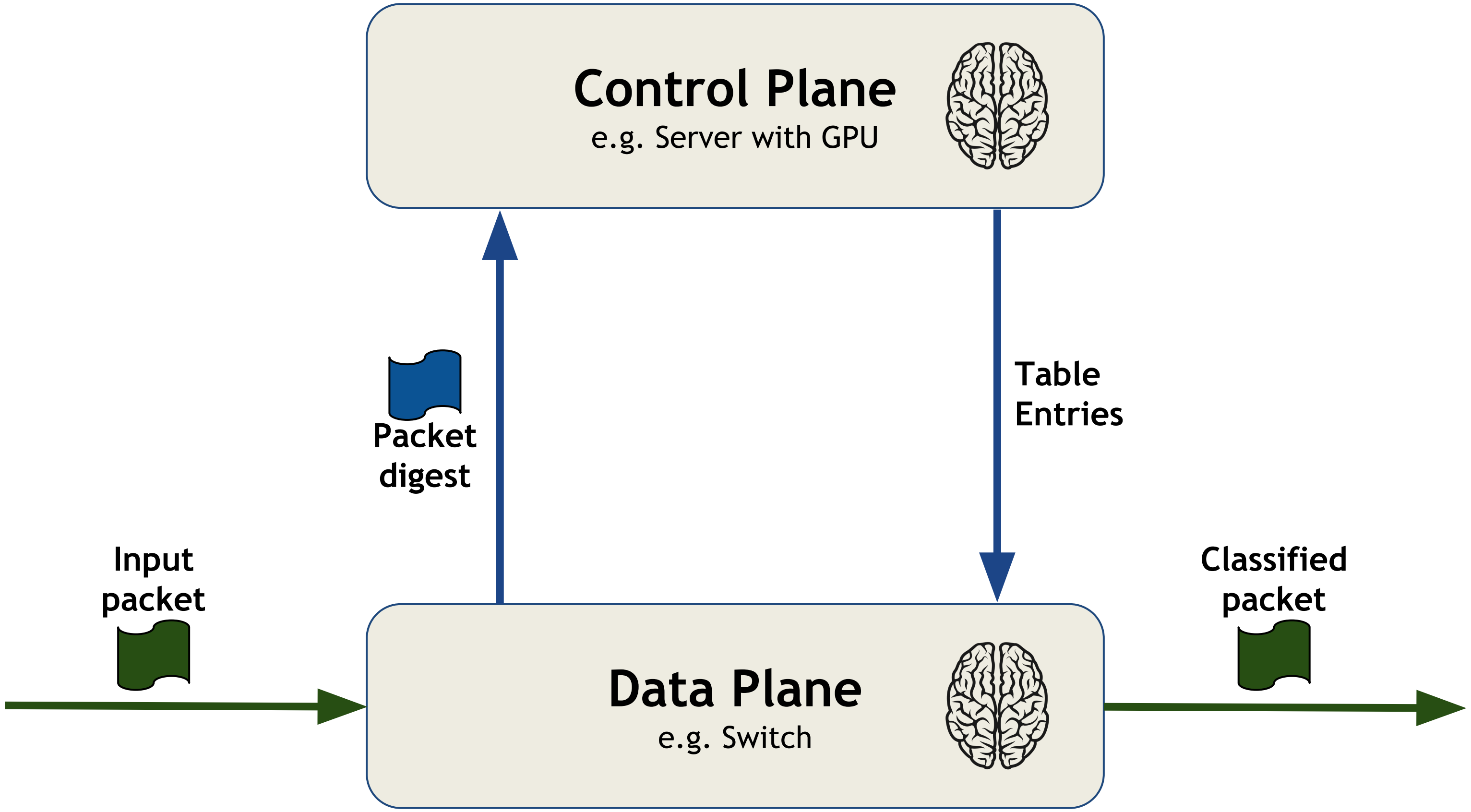
DDoS Attack Mitigation

Line-rate ML Inference

etc.

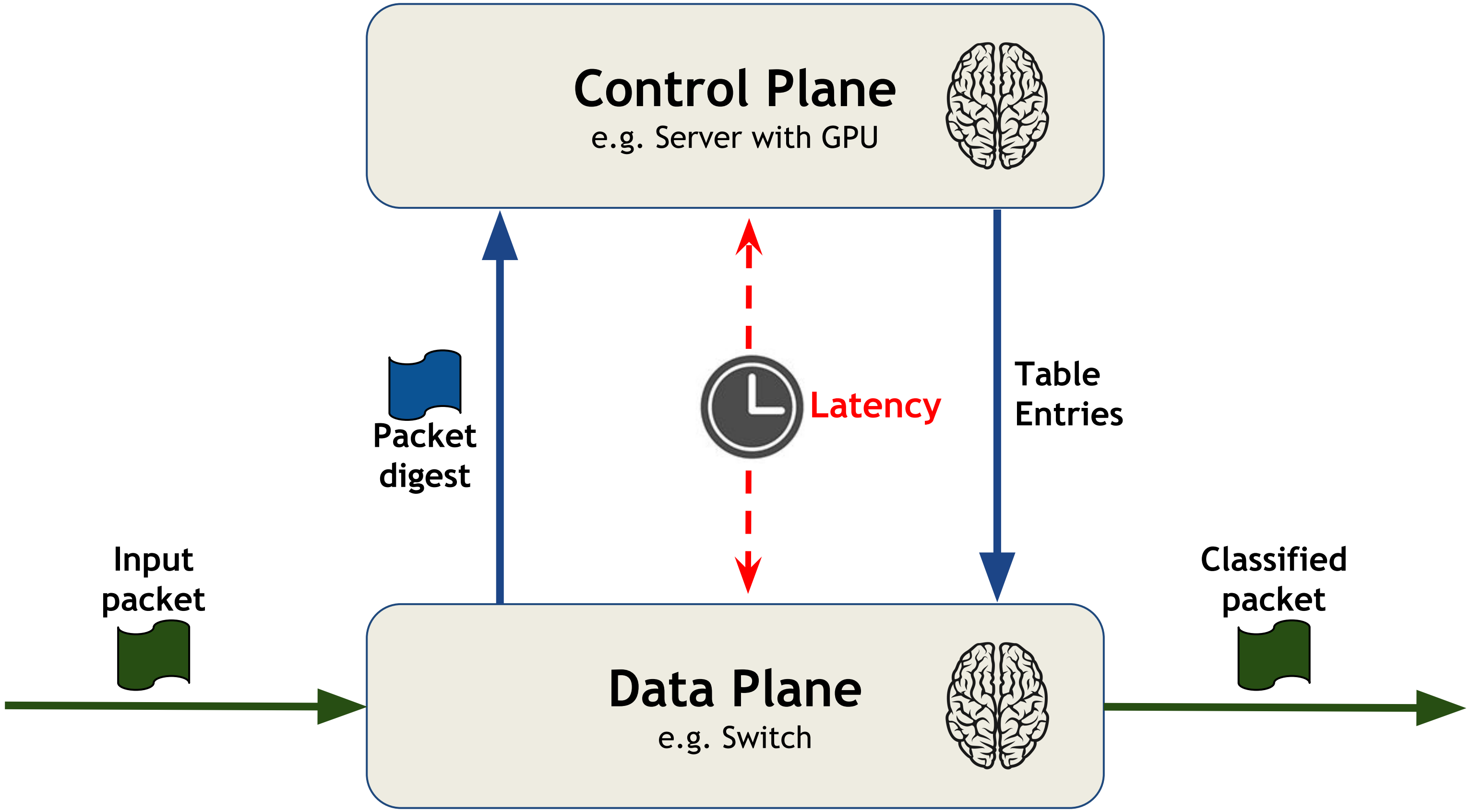
Machine Learning Inference in the Data Plane

Why Data Plane Machine Learning Inference?



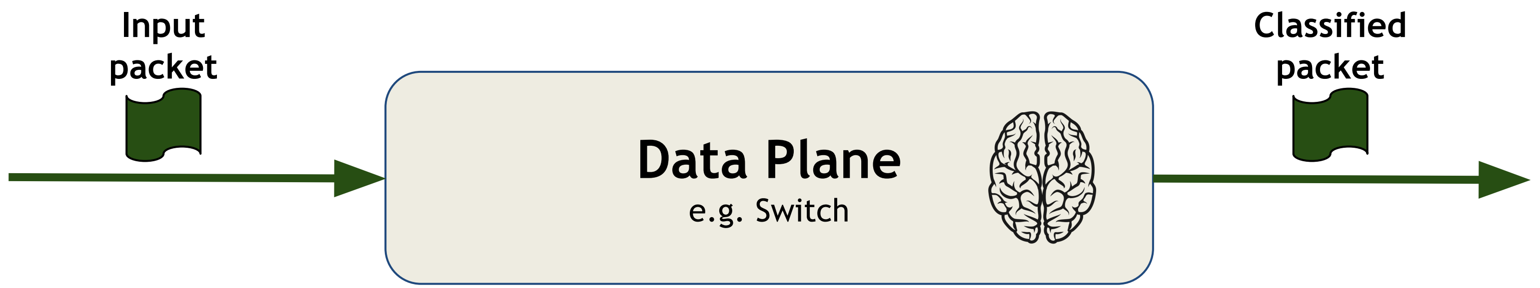
Traditional SDN

Why Data Plane Machine Learning Inference?



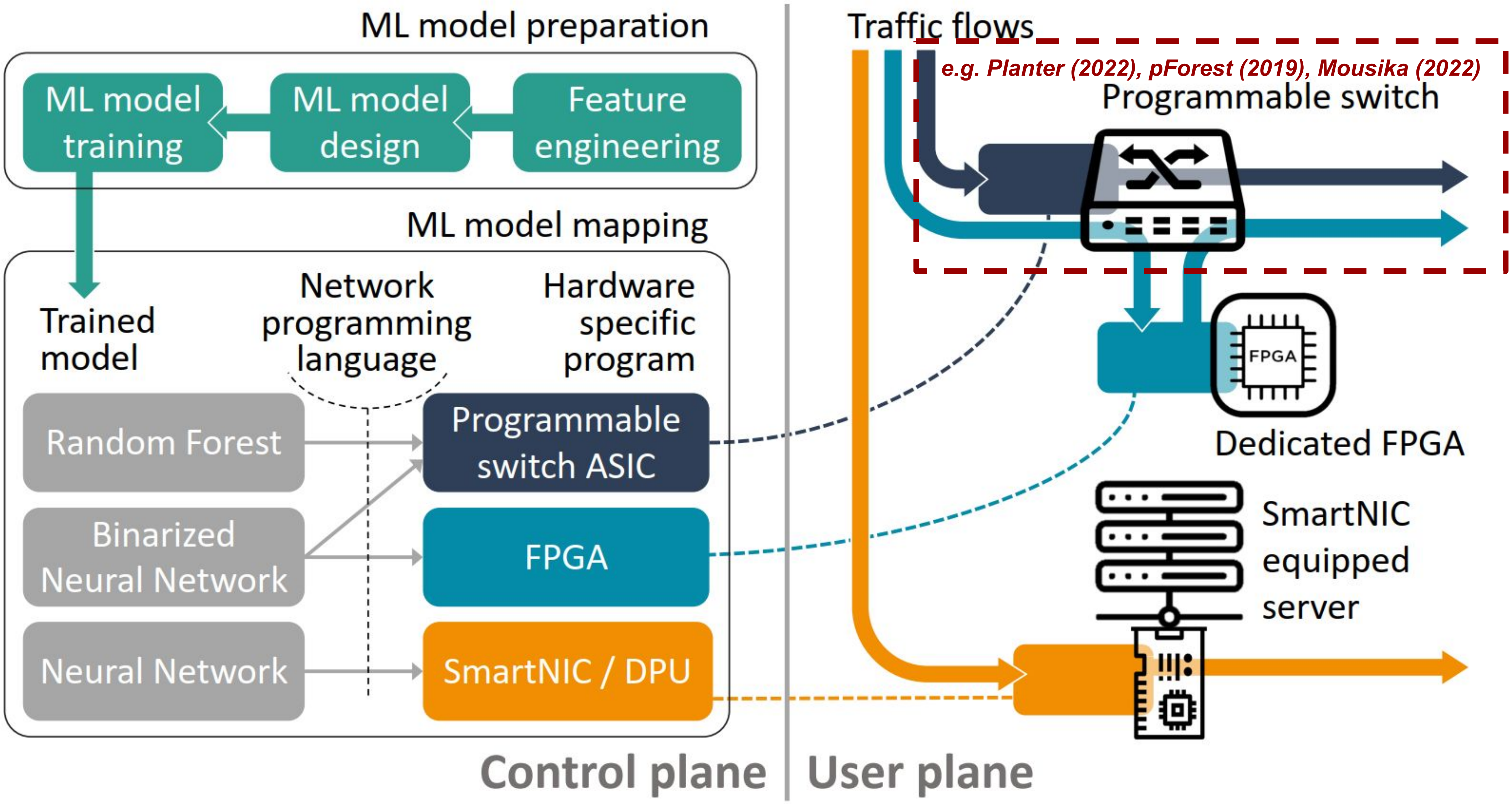
Communication with controller introduces a delay

Why Data Plane Machine Learning Inference?



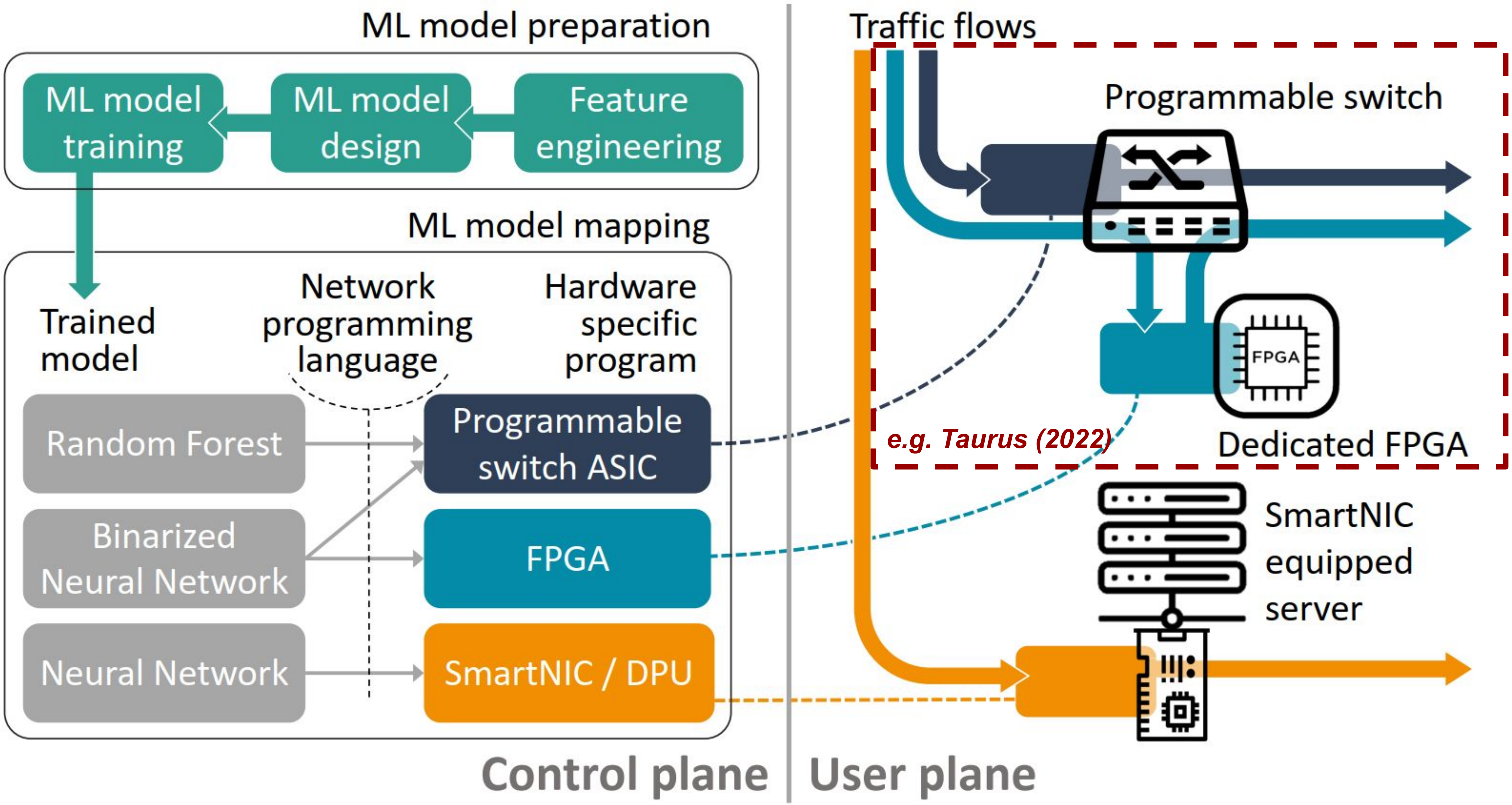
We want to eliminate the delay by bringing ML inference into the data plane

Line-rate Machine Learning Inference

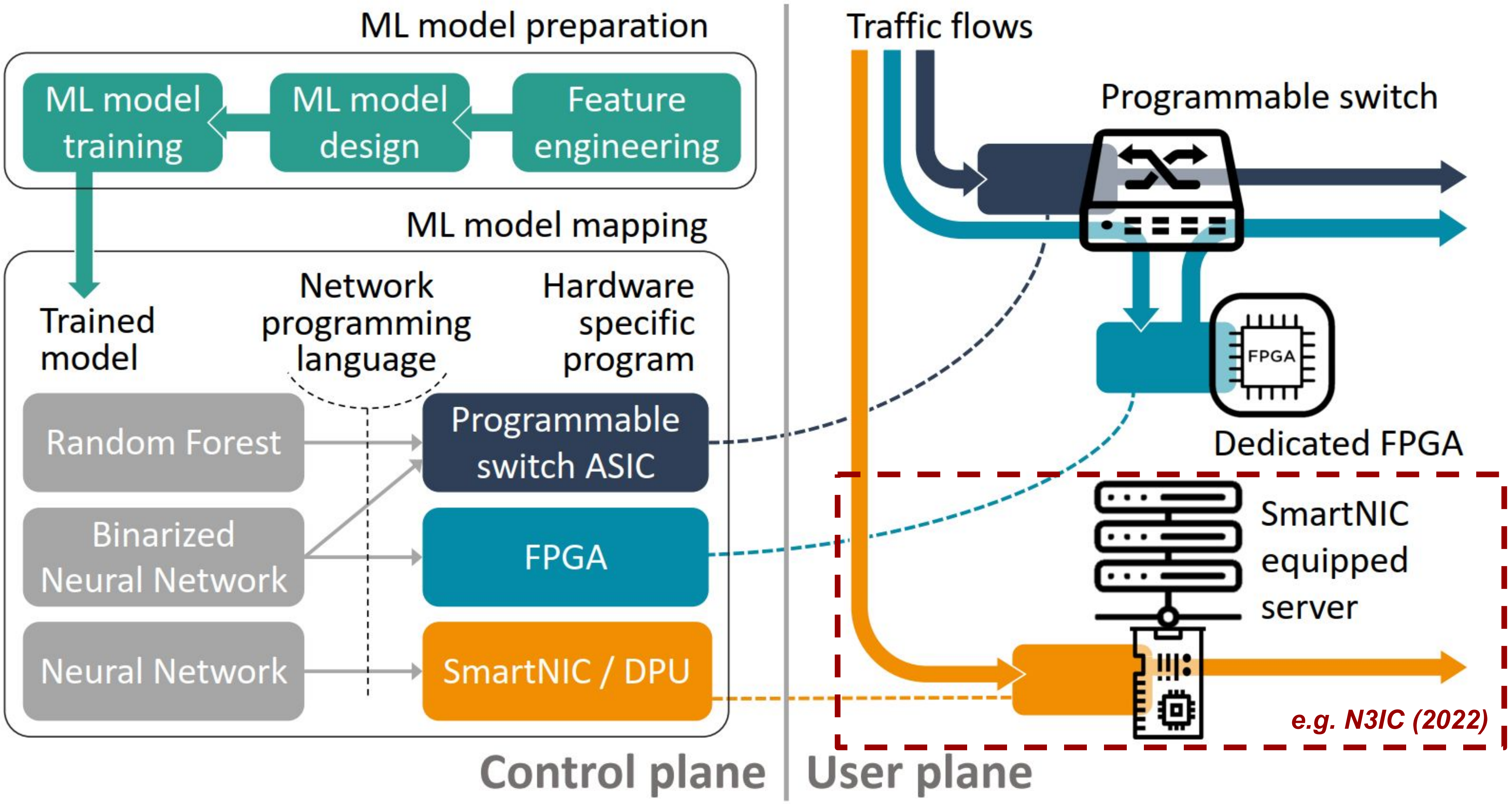


institute IMDEA networks

Line-rate Machine Learning Inference



Line-rate Machine Learning Inference



In-Switch Machine Learning Inference with Random Forests

State-of-the-Art

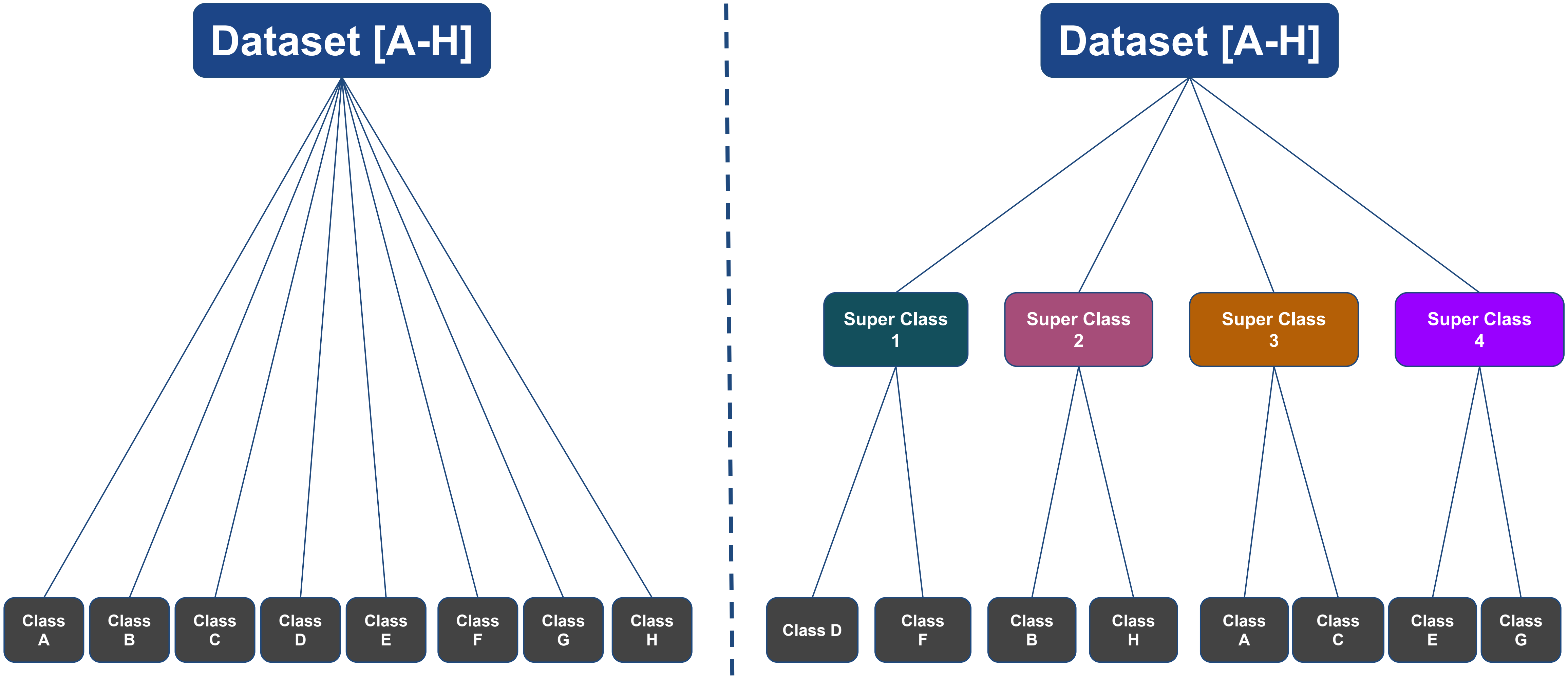
Problem → ***Train a single model for the problem and map it to the switch***

- IIsy *(Xiong et al, 2019)*
- pForest *(Busse-Grawitz et al, 2019 & 2022)*
- SwitchTree *(Lee et al, 2020)*
- Planter *(Zheng et al, 2021 & 2022)*
- NERDS *(Xavier et al, 2021)*
- pHeavy *(Zhang et al, 2021)*
- Mousika *(Xie et al, 2022)*

For difficult tasks, a monolithic classifier often becomes too complex to fit within switch resources while attaining the desired accuracy.

If hierarchical relationships exist, the task could be split into smaller tasks that are easier to solve, improve classification accuracy and fit within switch resources.

Hierarchical Machine Learning Inference



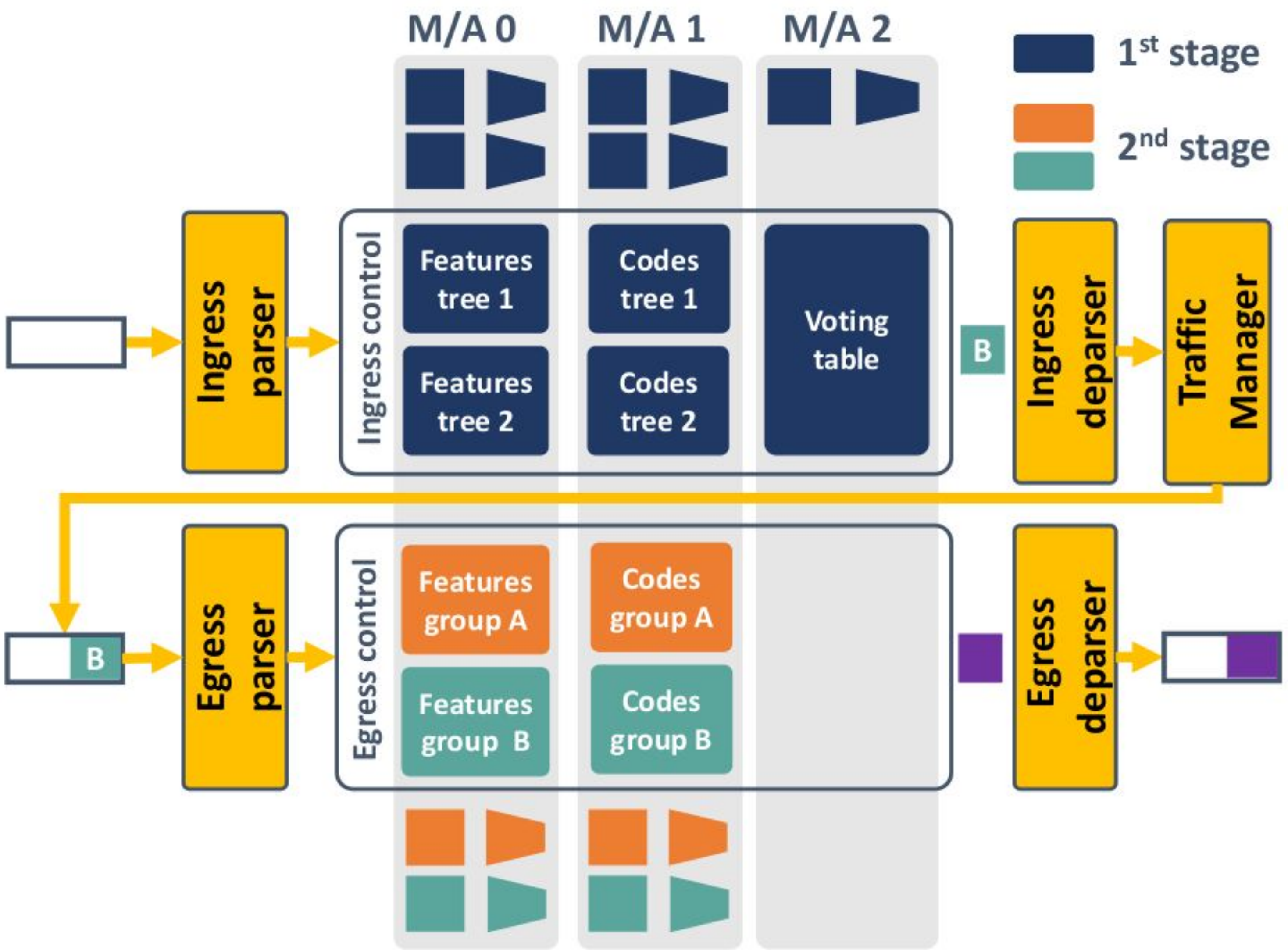
Henna: In-Switch Hierarchical Machine Learning Inference

Henna: Description

- **Implemented as a two-stage classifier**
  - *First stage - identify class groups, Second stage - identify individual classes*
- **First stage model in ingress, second stage models in egress**
  - *Allows models of both stages to share M/A stage resources in Tofino switch*
- **Models trained in Python using the Scikit-Learn library**
  - *Feature selection, grid search for hyperparameters, model validation & selection*
- **Henna uses the decision tree/random forest mapping in Planter[1]**
  - *Trees are mapped via feature range tables and tree code tables*

[1] Changgang Zheng and Noa Zilberman: Planter: seeding trees within switches. *SIGCOMM '21 Poster and Demo Sessions*. ACM, New York, NY, USA, 2021.

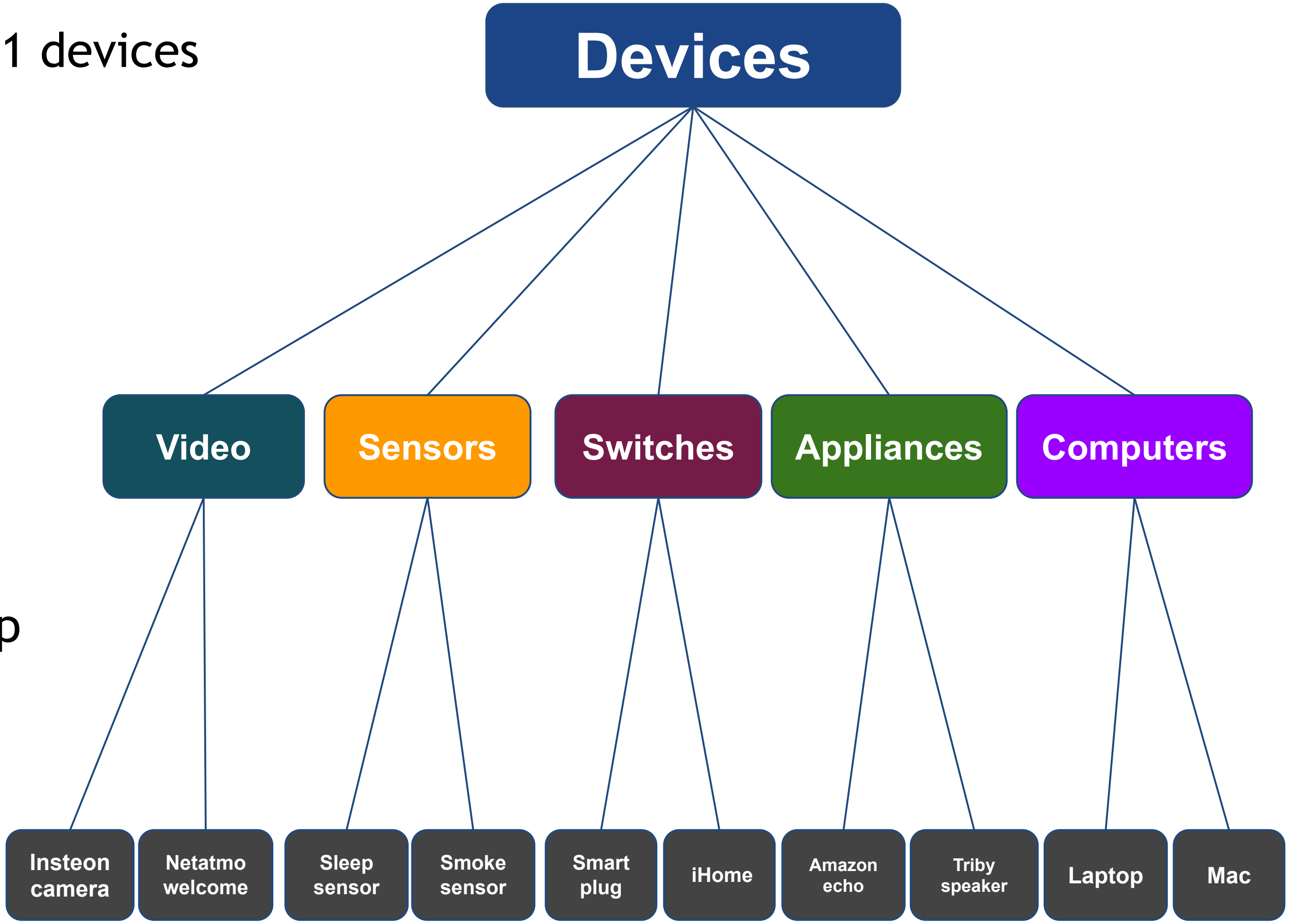
Henna Mapping into the PISA Architecture



Use Case and Experimental Setup

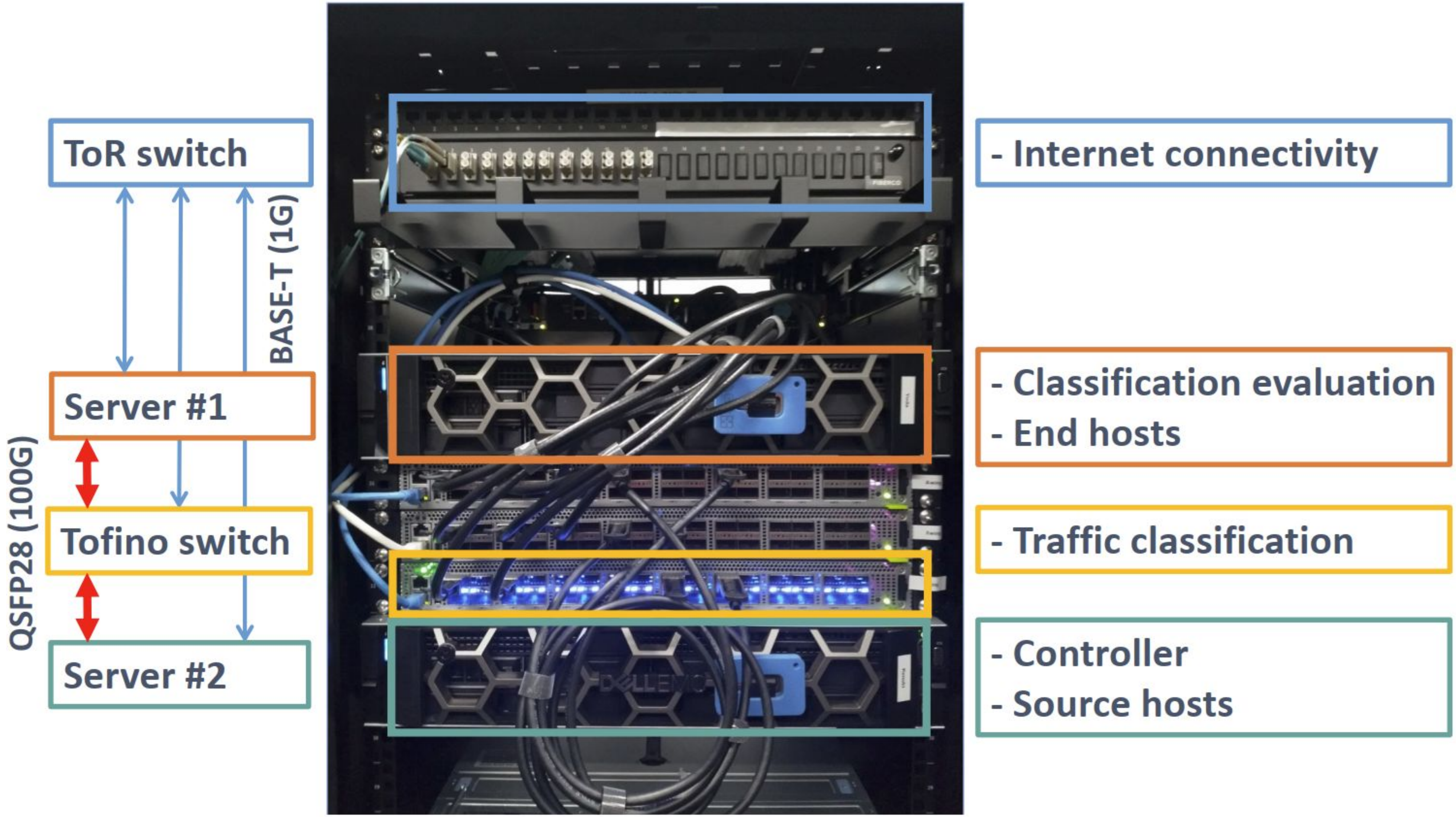
Use Case: UNSW-IoT Traces [1]

- Device Identification problem with 21 devices
- Devices clustered into 5 groups
    - Switches & Plugs
    - Sensors
    - Video Devices
    - Appliances
    - Computers
- **First stage:** identify the device group
- **Second stage:** identify the device
- **Benchmark:** single-stage classifier



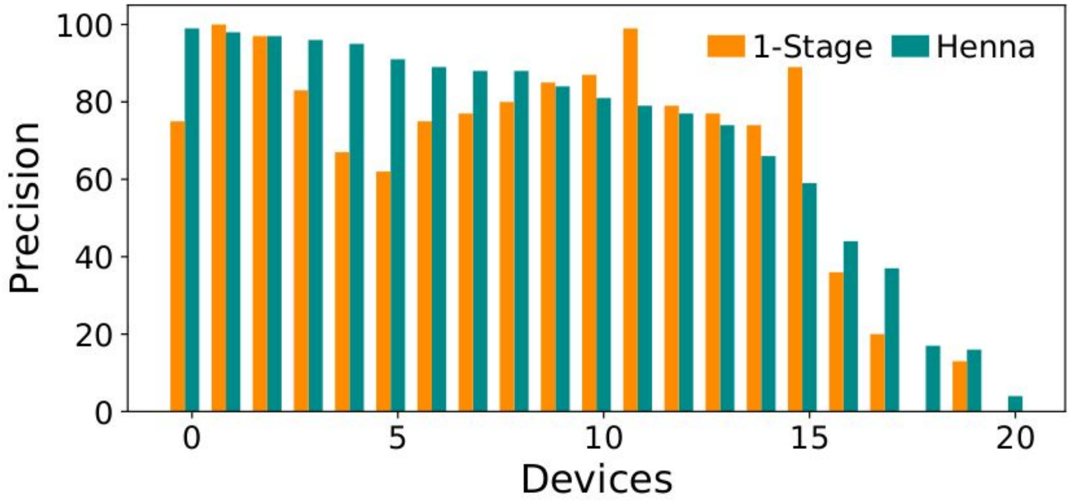
[1] https://iotanalytics.unsw.edu.au/iottraces.html

Experimental Setup

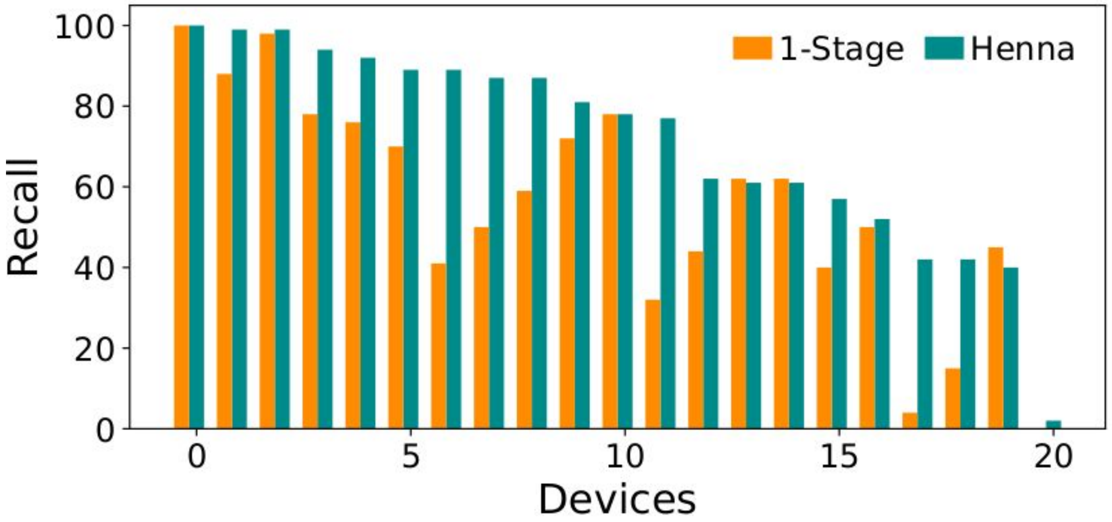


Results

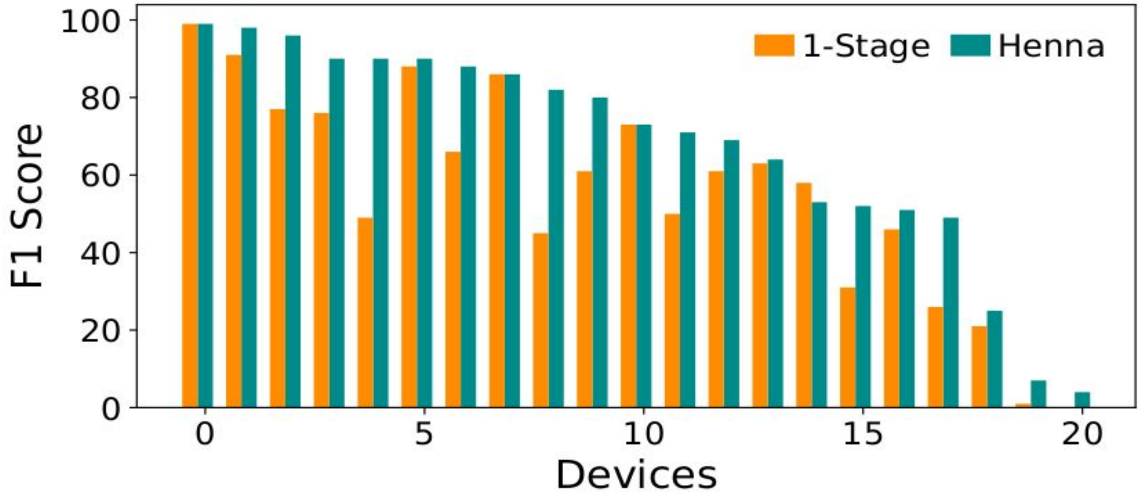
Results - Classification Accuracy



Precision = TP / TP + FP



Recall = TP / TP + FN



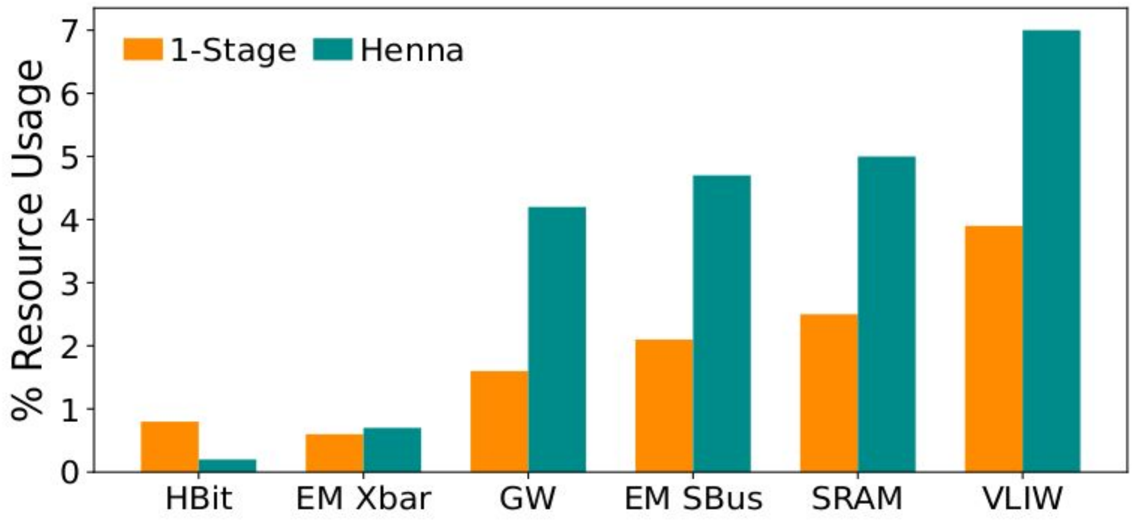
F1 Score = 2 x (Precision x Recall)/(Precision + Recall)

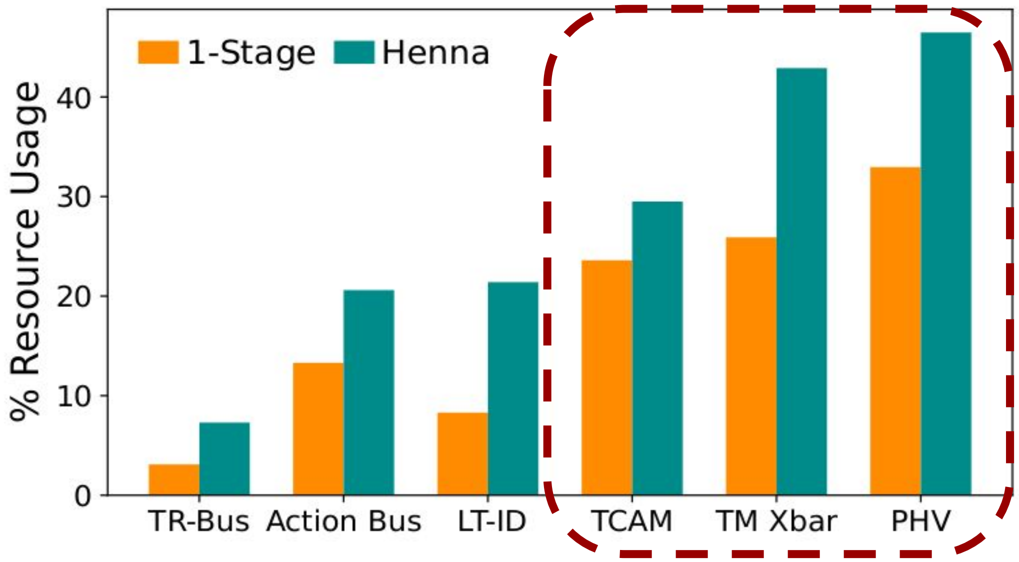
Gain in F1 score:

- Absolute: **11.95%**
- Relative: **21.52%**

Results - Resource Usage

On average Henna consumes about 8% of total switch resources





Significant use of TCAM and PHV memory (inherent to RF mapping used)

Conclusion

Conclusion

- We presented Henna, a two-stage decision tree-based in-switch packet classifier
- We implemented our solution in a real-world experimental platform
- Results show that Henna improves classification performance with respect to a monolithic classifier while keeping resource usage under control.
- Future work will seek to extend Henna to flow classification, reduce resource consumption and explore new use cases.

Code available: https://github.com/nds-group/Henna

Thank you!

DAEMON

BANYAN

MINISTERIO
DE CIENCIA
E INNOVACIÓN

This work has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement no. 101017109 “DAEMON”, and the Marie Skłodowska-Curie grant agreement no. 860239 “BANYAN”.