

# Planter: Seeding Trees Within Switches

Changgang Zheng

✉ [changgang.zheng@eng.ox.ac.uk](mailto:changgang.zheng@eng.ox.ac.uk)

Research Advisor: Prof. Noa Zilberman

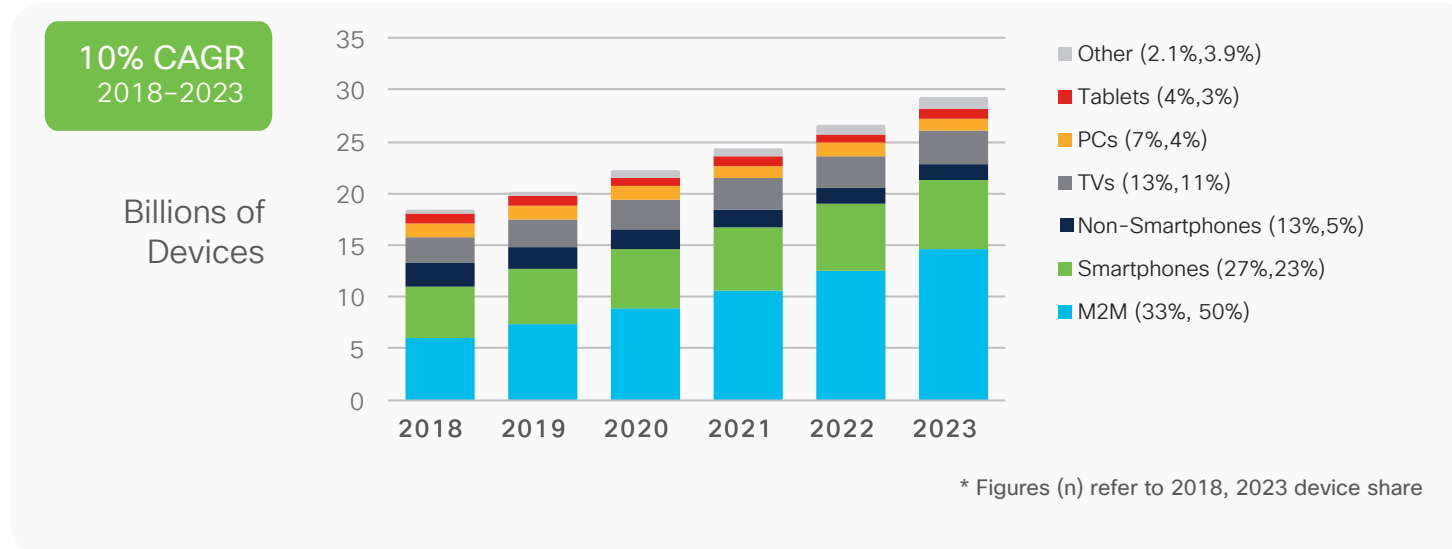
Joint work with Zhaoqi Xiong, Siim Kaupmees, Yaniv Ben-Itzhak and Shay Vargaftik

coseners2021 - 1st July 2021

# Why on-switch classification?

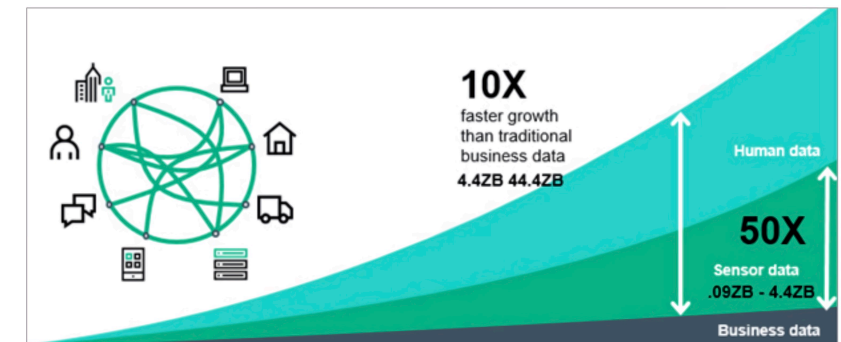
## Growth of network connections

Figure 2. Global device and connection growth



Source: Cisco Annual Internet Report, 2018-2023

## Exponential growth of data



Source: The Intelligent Use of Big Data on an Industrial Scale

# Why on-switch classification?

## Anomaly detection use case

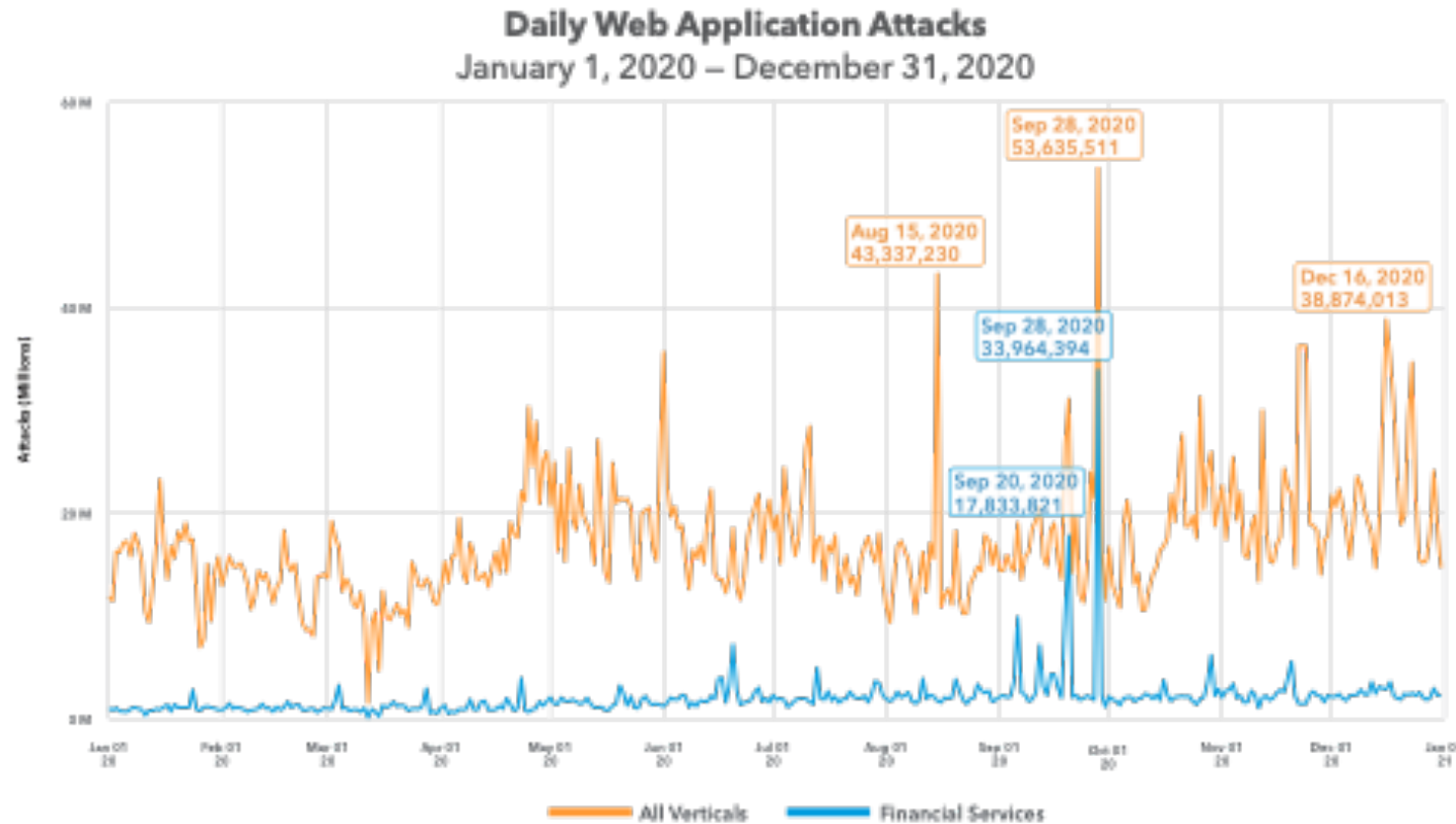
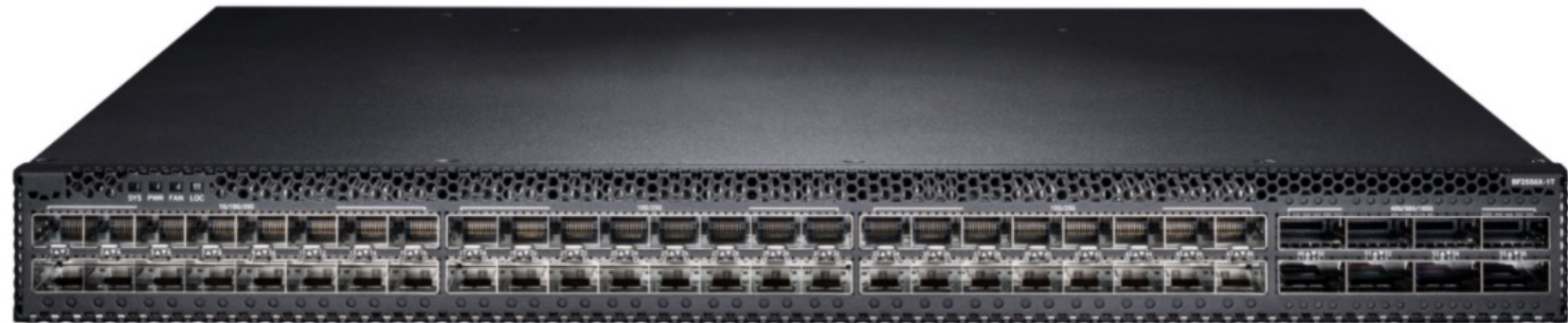


Fig. 2: Web attacks reached five notable peaks in 2020, all in Q3 and Q4

Source: 2021 State of the Internet / Security: Phishing for Finance Phishing for Finance

# Why on-switch classification?

1. Better energy efficiency
2. Processing speed

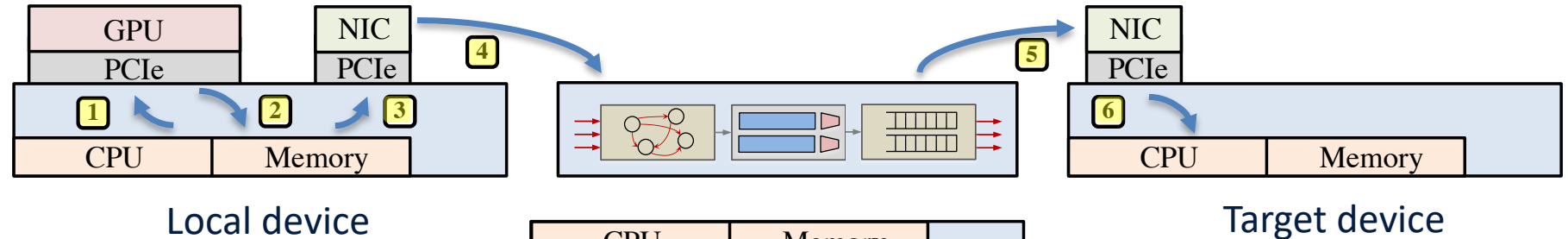


Intel® Tofino™ 6.4 Tbps 64x100Gb switch.

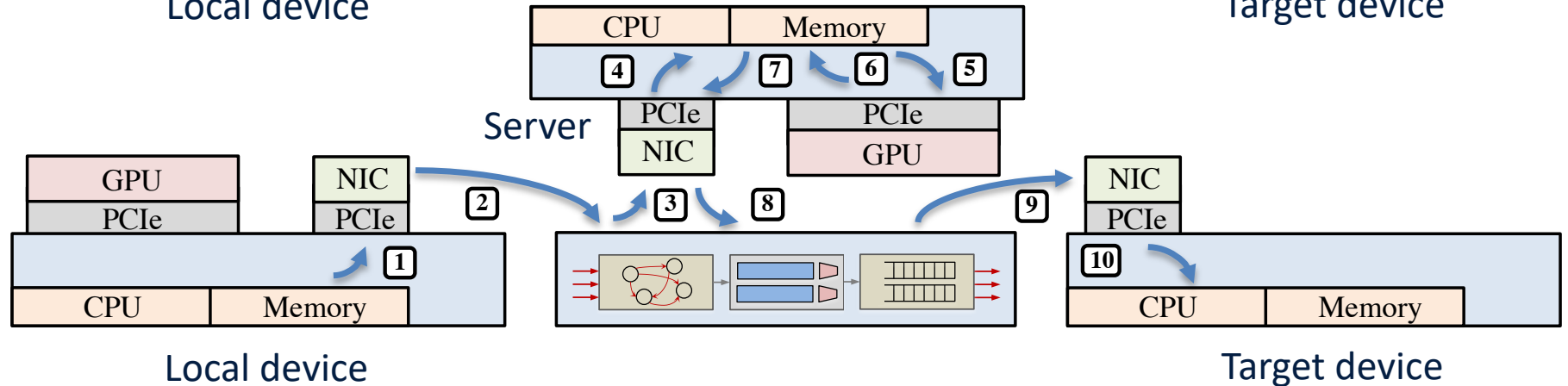
# Why on-switch classification?

## Better Position

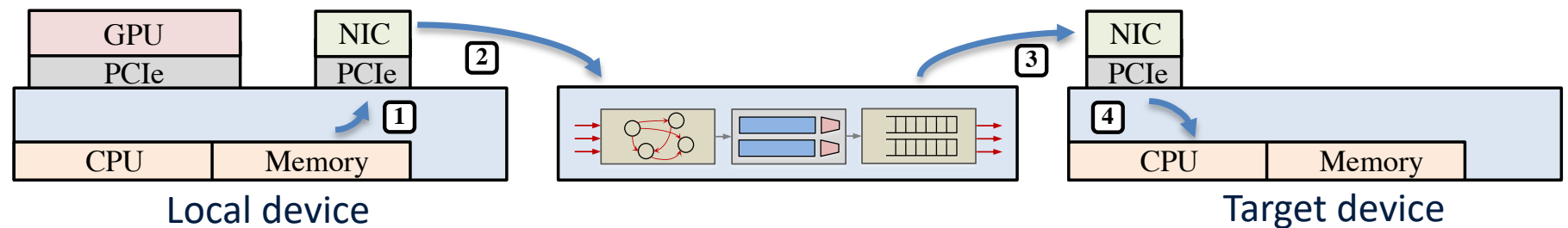
On local GPU:



On Server GPU:



On Switch:



# On-switch classification challenges

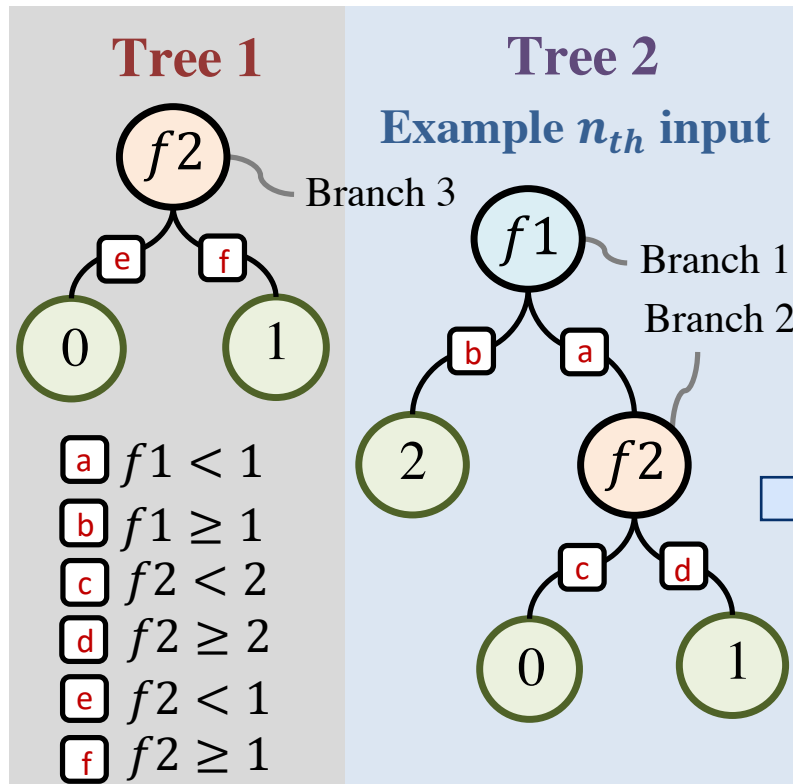
1. Limited mathematical operations
2. Limited memory
3. Limited data types
4. Limited stages

The **APS Networks BF6064X-T** is a dedicated platform for intelligent network programming, SDN and NFV development. Capture the full benefits of the powerful Intel® Tofino™ 6.4 Tbps 64x100Gb switch.



Source from: Advanced Programmable Switch [www.aps-networks.com](http://www.aps-networks.com)

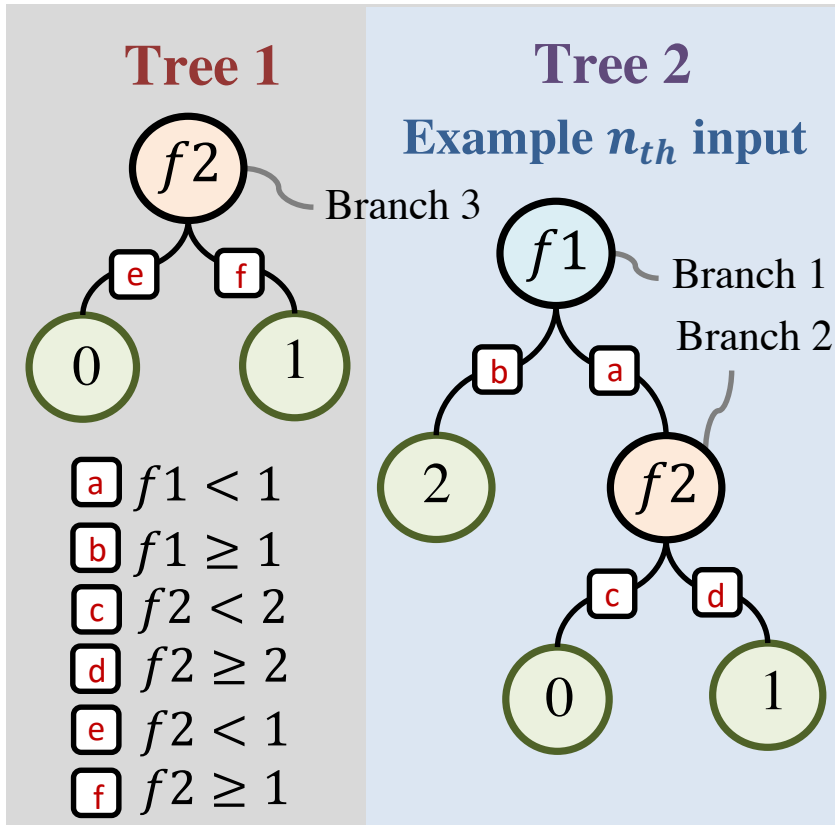
# Our goal



The **APS Networks BF6064X-T** is a dedicated platform for intelligent network programming, SDN and NFV development. Capture the full benefits of the powerful Intel® Tofino™ 6.4 Tbps 64x100Gb switch.



# What is & Why ensemble models?



From machine learning perspective

1. Robust.
2. Straight forward and explainable.
3. Adapt to different learning tasks.
4. Widely used.

From implementation perspective

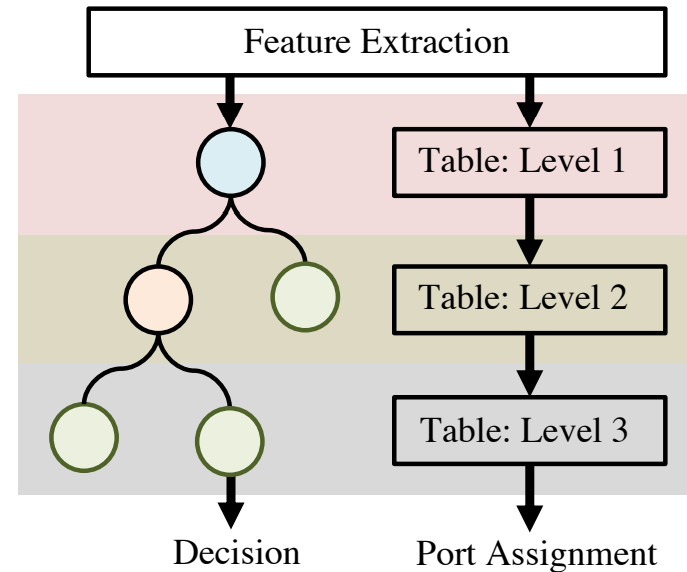
1. Simple structure and operation



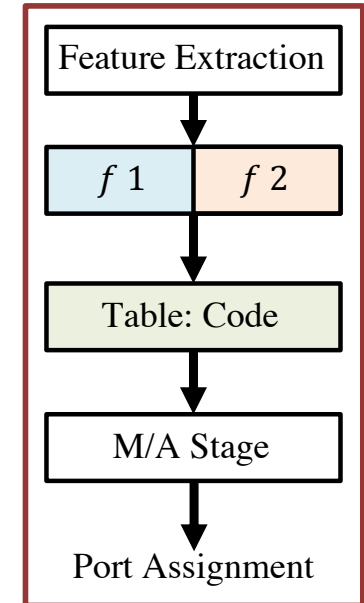
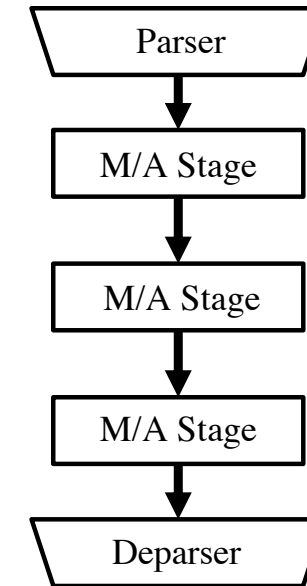
# Two mapping approaches

**SwitchTree & pForest** use a match-action stage for each level in the tree.

**Ilsy & Planter** encode the model using a table per feature, coding the decision into tree leaves.



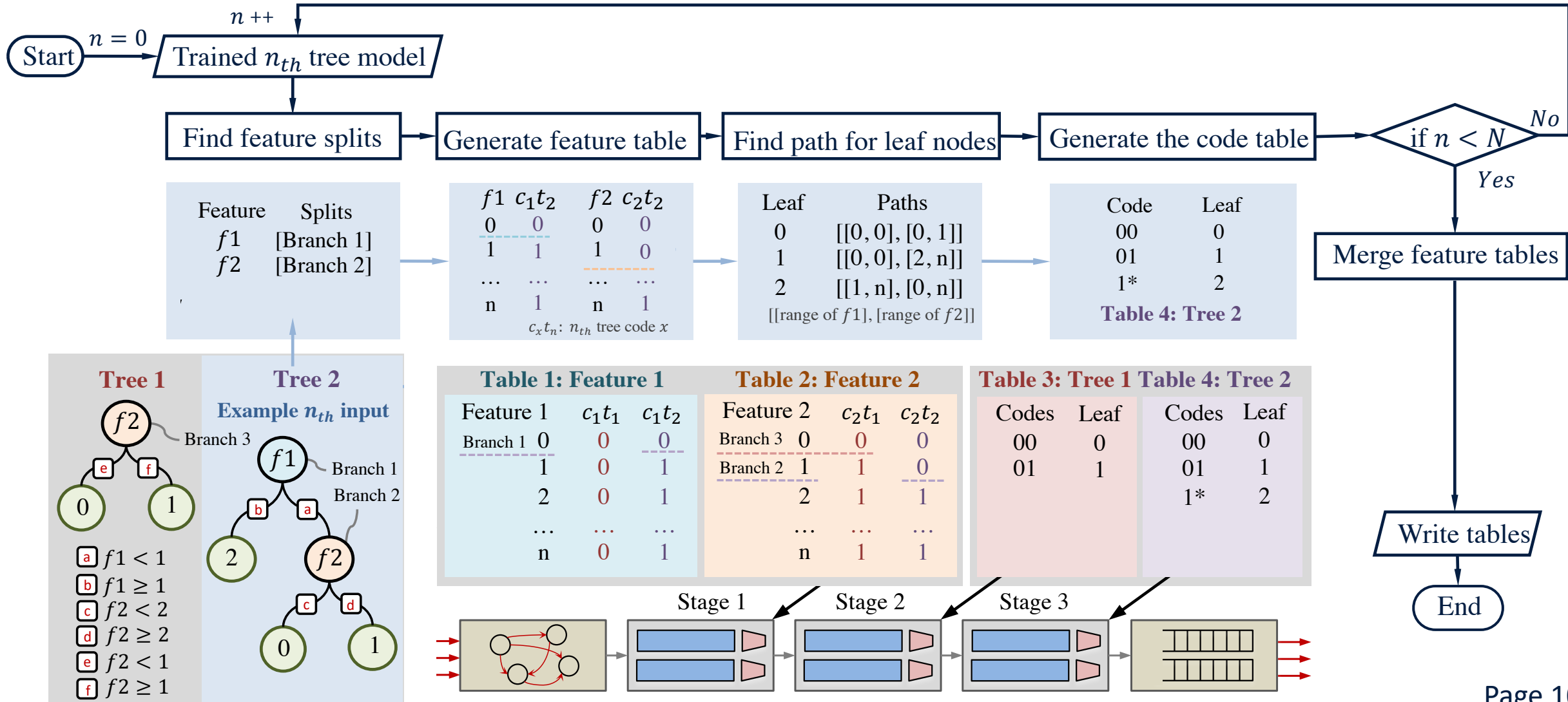
(a) SwitchTree & pForest



(b) Ilsy & Planter

Planter: the number of stages is independent of model's depth.

# Planter algorithm



# Some ensemble models

## XGBoost

$$label = \arg \max_x (p_x)$$

$$p_1 = prob_1 + \dots + prob_{j-1}$$

$$p_2 = prob_j + \dots + prob_k$$

Table  $m + 1$  to Table  $m + k$ : Code to Probability

Code	Tree 1	...	Code	Tree k
$codes_1$	$prob_1$	...	$codes_1$	$prob_k$
...	...	...	...	...
$codes_n$	$prob_1$	...	$codes_n$	$prob_k$

Tree  $i$  codes =  $code_{i1} ++ code_{i2} \dots ++ code_{im}$

## Random Forest

$$label = \arg \max_x (label_x)$$

$$label_1 = vote_1 + \dots + vote_{j-1}$$

$$label_k = vote_j + \dots + vote_k$$

Table  $m + 1$  to Table  $m + k$ : Code to Probability

Code	Tree 1	...	Code	Tree k
$codes_1$	$vote_1$	...	$codes_1$	$vote_k$
...	...	...	...	...
$codes_n$	$vote_1$	...	$codes_n$	$vote_k$

Tree  $i$  codes =  $code_{i1} ++ code_{i2} \dots ++ code_{im}$

## Isolation Forest

if  $depth > threshold$   
label = anomaly

$$depth = depth_1 + \dots + depth_k$$

Table  $m + 1$  to Table  $m + k$ : Code to Depth

Code	Tree 1	...	Code	Tree k
$codes_1$	$depth_1$	...	$codes_1$	$depth_k$
...	...	...	...	...
$codes_n$	$depth_1$	...	$codes_n$	$depth_k$

Tree  $i$  codes =  $code_{i1} ++ code_{i2} \dots ++ code_{im}$

Table 1: Feature 1 to code

Key	Tree 1	...	Tree k
$k_1$	$code_{11}$	...	$code_{k1}$
...	...	...	...
$k_n$	$code_{11}$	...	$code_{k1}$

Table m: Feature m to code

Key	Tree 1	...	Tree k
$k_1$	$code_{1m}$	...	$code_{km}$
...	...	...	...
$k_n$	$code_{1m}$	...	$code_{km}$

# Preliminary evaluation on Tofino

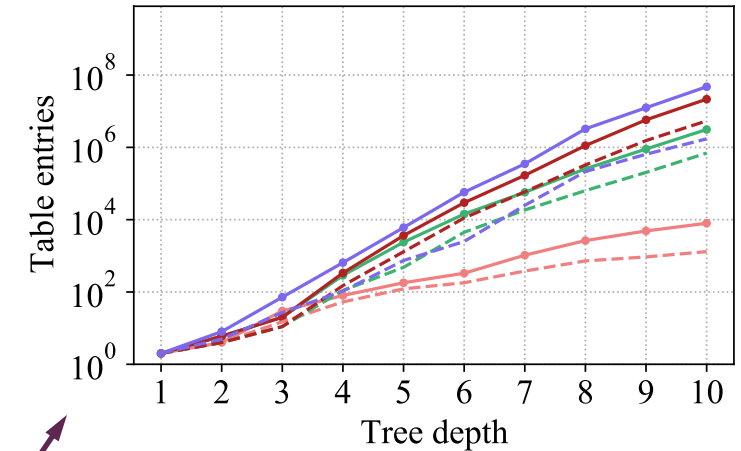
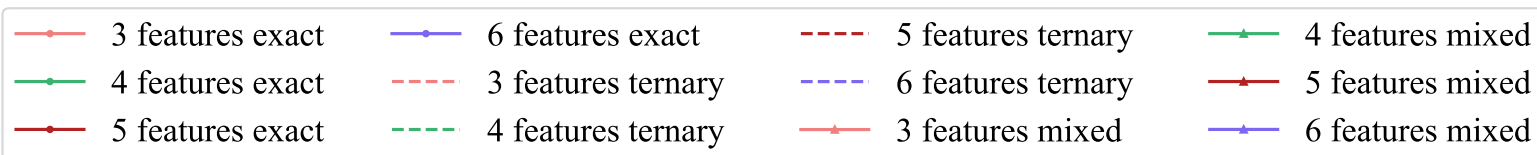
## Flowers classification (Iris dataset)

Model	Trees	Tables	Memory	Latency	Acc	Baseline
RF	3	7	1.4%	32.5%	97.8%	97.8%
XGB	3	8	1.3%	41.1%	97.8%	97.8%

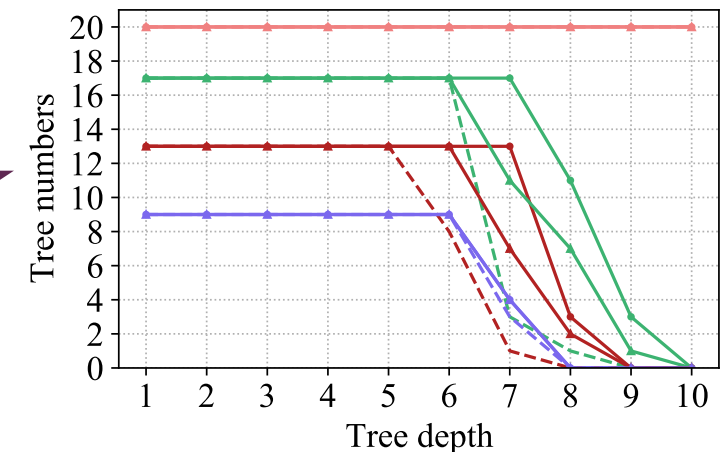
## Anomaly detection (UNSW dataset)

Model	Trees	Tables	Memory	Latency	Acc	Baseline
RF	6	11	6.8%	64.0%	97.3%	98.5%
XGB	6	11	6.7%	64.3%	96.7%	98.7%

## Resource consumption of anomaly detection use case

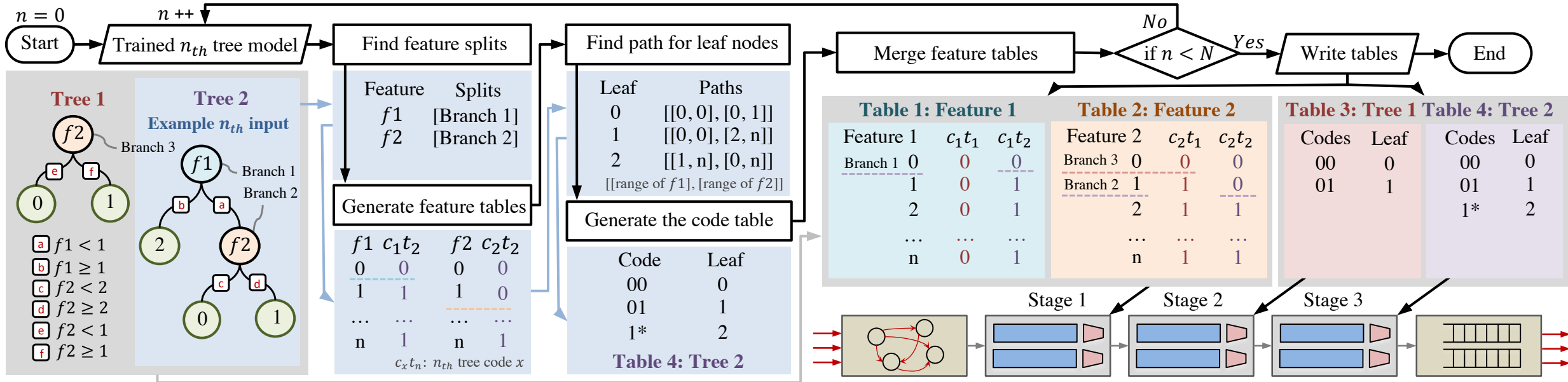


Depth - Table entries



Depth - Tree numbers

# Planter: Seeding Trees Within Switches



Changgang Zheng

✉ changgang.zheng@eng.ox.ac.uk