DNN Model Architecture Fingerprinting Attack on CPU-GPU Edge Devices

Kartik Patwari, Syed Mahbub Hafiz, Han Wang,
Houman Homayoun, Zubair Shafiq, Chen-Nee Chuah

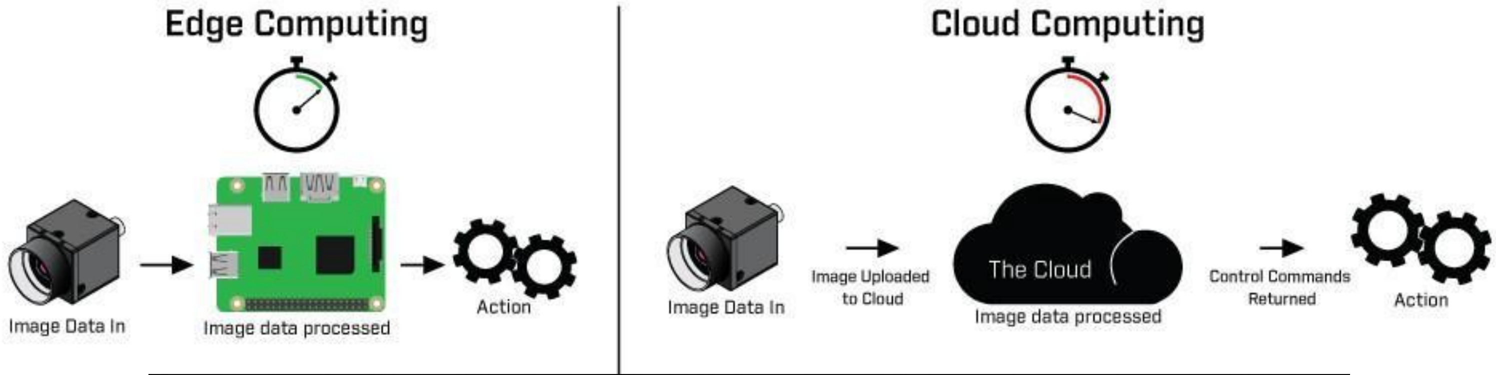
June 8, 2022

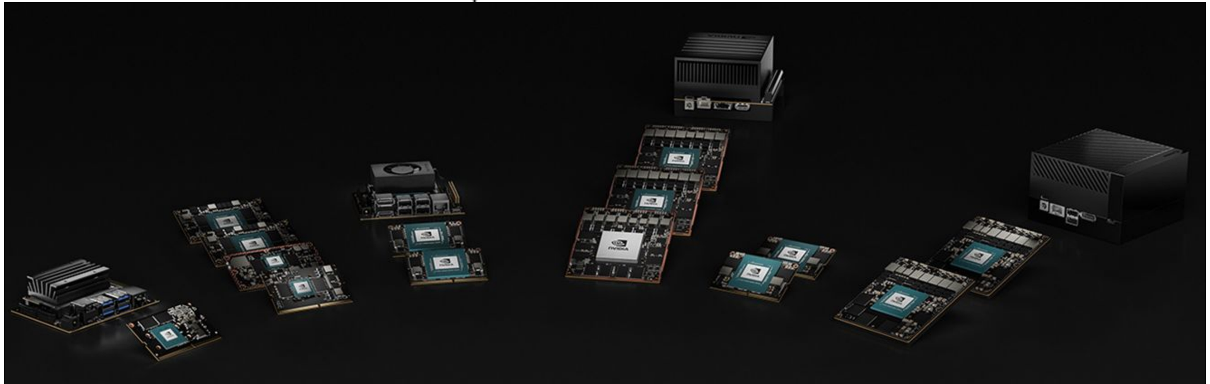
EuroS&P

UCDAVIS
ELECTRICAL AND COMPUTER ENGINEERING

UCDAVIS ENGINEERING

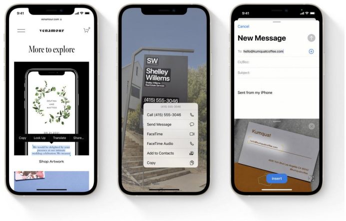
Motivation: Deep Learning on Edge Devices





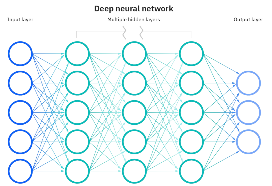
Model Extraction Attacks

Black-Box Setting



Side-Channel Attacks (SCAs)

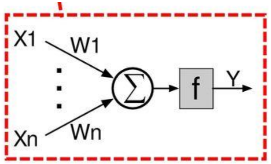
White or Grey-Box Knowledge



Architecture

CNN	STFT	Conv layer	1-2-5; 2-3-5; 3-3-5	3
		Max pool layer	2-2	1
		Fully connection layer	1980-32; 32-5	2
Dropout	**Optimizer**	**Training Epochs**	**Learning Rate**	**Batch Size**
0.5	Adam	100	0.02	15

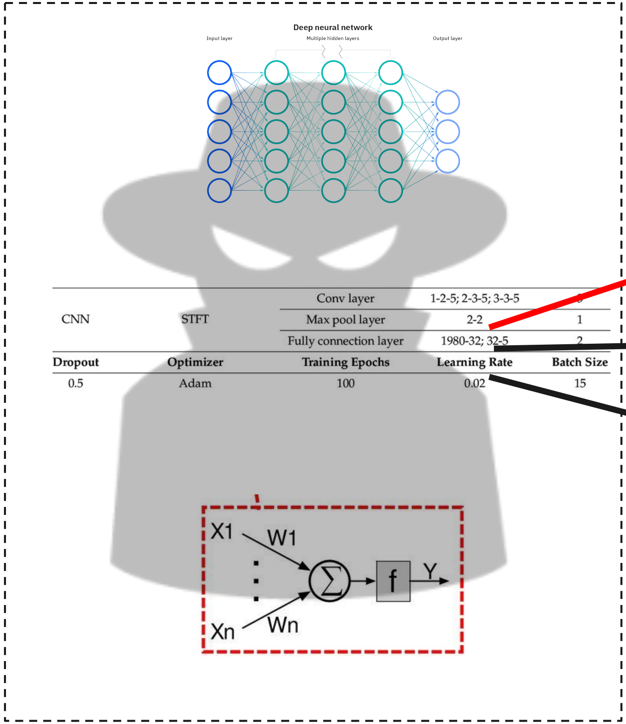
Hyper-parameters



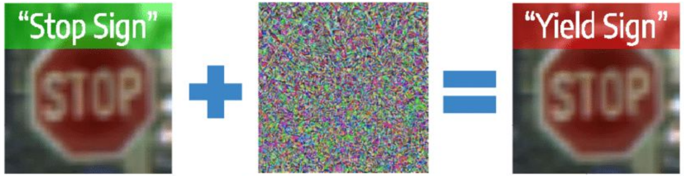
Weights

White/Grey-Box Knowledge is Useful

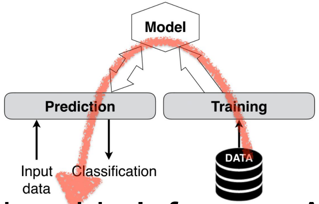
White or Grey-Box



Security & Privacy Threats!



Adversarial Attacks



Membership Inference Attacks

Model Inversion Attacks

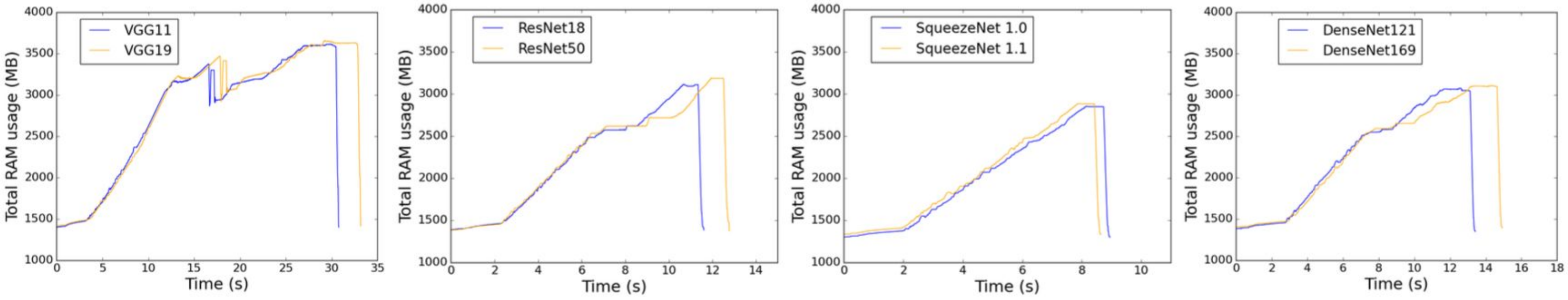
Comparison with Prior Work

SCA classification - (1) *Invasive vs.* ***Non-Invasive*** (2) *Active vs.* ***Passive*** *(3) Remote*

Attack	**Side-Channel**	**Classification**	**Limitation(s)**
Cache Telepathy (USENIX ‘20)	Cache	Non-invasive, semi-active, remote	Requires executing Prime+Probe attack code & LLC sharing
DeepSniffer (ASPLOS ‘20)	Memory Access	Semi-invasive, passive	Physical Access required, Bus snooping
Leaky DNN (DSN ‘20)	GPU	Non-invasive, semi-active, remote	Cloud GPU based, profilers required, attack code & DoS attack needed
CSI NN (USENIX ‘19)	Power/EM	Non-invasive, passive	Physical Access required
Our Work (EuroS&P ‘22)	**Aggregate System-level Statistics**	**Non-invasive, passive, remote**	**None of the above**

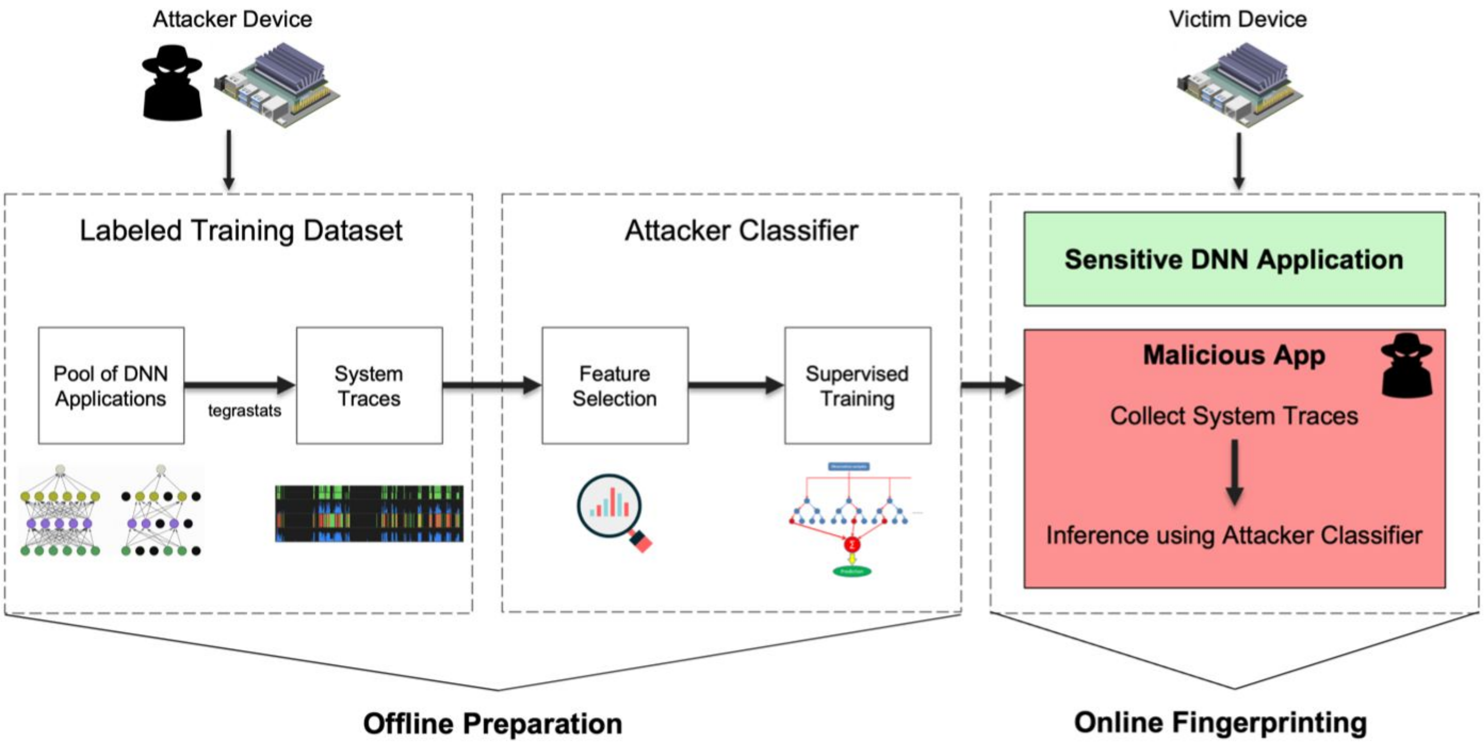
Threat Model

- **Attacker's Goal**
  - **Fingerprint Model Architecture Family** from popular, state-of-the-art DNNs
  - Model architecture family knowledge **improves black-box ensemble adversarial attacks**
- **Attacker's Knowledge**
  - The victim device to have a **clone attacker device**
  - Victim DNN belongs to one of **known DNN families**, primary running application
- **Attacker's Capability**
  - Able to collect **global system-level** statistics available at **user-space level** (e.g. *tegrastats* for Jetson devices)
  - Total **RAM** usage, and **CPU**, **GPU** load(s)



(a) VGG Family (b) ResNet Family (c) SqueezeNet Family (d) DenseNet Family

Attack Pipeline

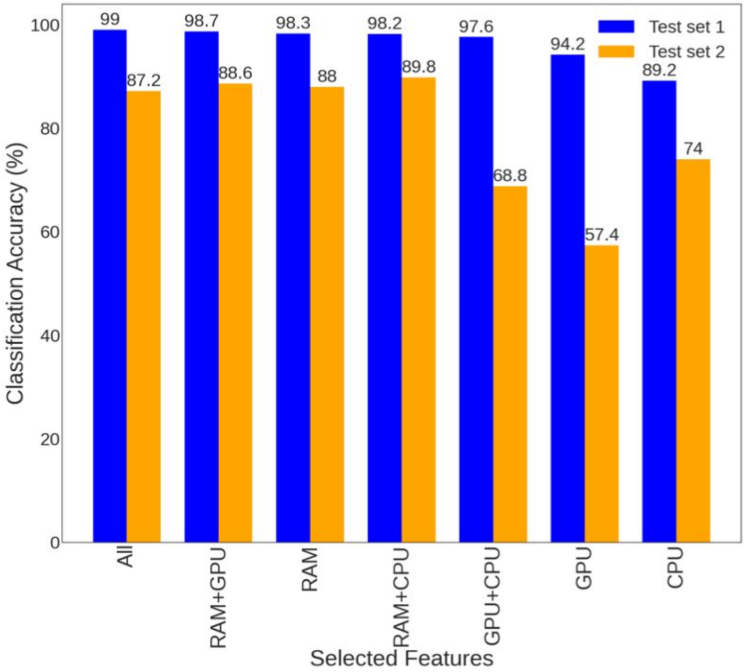


Experimentation Setup

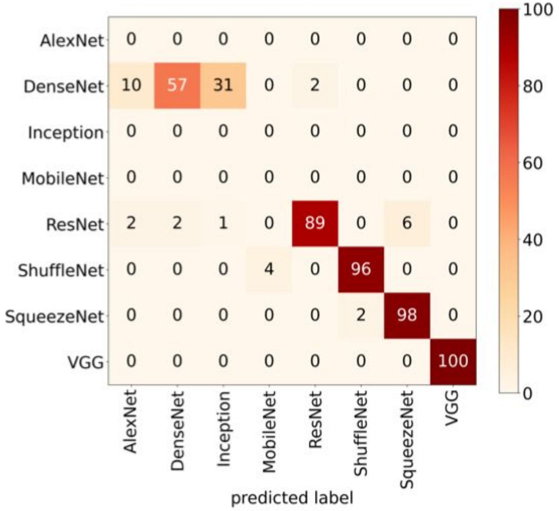
- Edge Testbed – NVIDIA Jetson Devices
  - GPU-enabled edge AI devices with unified memory
  - Jetson Nano (4GB)
    - 4-core ARM Cortex A57, 128-Core Maxwell, 4GB Memory
- Global statistics from NVIDIA *tegrastats*:
  - Total RAM consumption
  - Aggregate CPU Load(s)
  - Aggregate GPU Load
- All pretrained models obtained from *torchvision*
- Test set 2 is for evaluating *transferability* of the attack

Model family	Train/Test set 1	Test set 2
VGG	VGG11, 19	VGG13, 16
ResNet	ResNet18, 50, 152	ResNet34, 101
SqueezeNet	SqueezeNet 1.0	SqueezeNet 1.1
DenseNet	DenseNet121, 201	DenseNet161, 169
ShuffleNet	ShuffleNetv2 0.5	ShuffleNetv2 1.0
Inception	InceptionV3	N/A
MobileNet	MobileNetv2	N/A
AlexNet	AlexNet	N/A

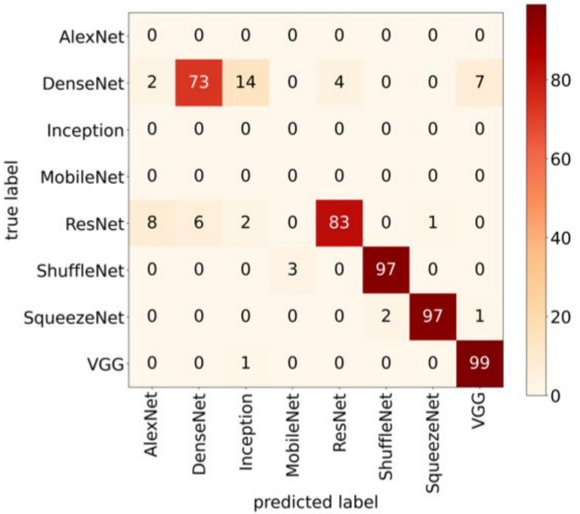
Feature Ablation and Transferability Study



Test set 2 Confusion Matrices:



(b) RAM only



(c) RAM and CPU

*All results are on Jetson Nano (4GB)

Enhancing Adversarial Attacks

- Ensemble Adversarial Attack examples from DeepFool [1] attack
- Three scenarios of ensemble example generation:
  - Models from same family (excluding victim model family)
  - Random Mix of models and families
  - Victim Model family

Adversarial examples generated by DeepFool on the ensemble of	Classification accuracy of victim model DenseNet121		Accuracy drop (%)
	W/o adv. perturbation (%)	W/ adv. perturbation (%)	
ResNets (ResNet50, 101, 152)	84.14	61.02	23.12
MobileNets (MobileNet, V2)	83.43	59.46	23.97
VGGs (VGG16, 19)	82.73	51.07	31.66
Mix 1 (MobileNet, ResNet50, EfficientNet)	83.85	63.9	19.95
Mix 2 (MobileNet, VGG16, EfficientNet)	83.69	58.08	25.61
Mix 3 (MobileNet, DenseNet121, EfficientNet)	83.72	53.55	30.17
Mix 4 (ResNet152, MobileNetV2, DenseNet201)	83.07	52.02	31.05
DenseNets (DenseNet121, 169, 201)	83.13	28.23	54.9

[1] "Deepfool: a simple and accurate method to fool deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Platform Portability

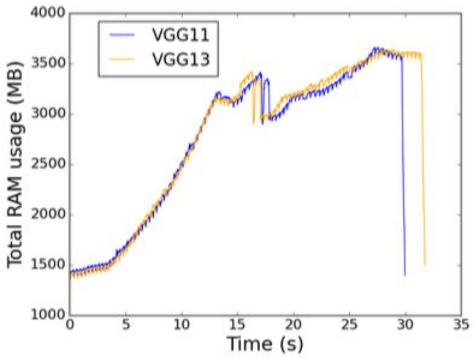
- Jetson TX2
  - 4-core Cortex ARM A57 + 2-core NVIDIA Denver 2, 256-core Pascal, 8GB Memory
- Jetson Xavier NX
  - 6-core NVIDIA Carmel ARM, 384-core Volta, 8GB Memory

Dataset \ Features	All	RAM	GPU	CPU	RAM+GPU	RAM+CPU	GPU+CPU
NX Test set 1	98.5%	98.6%	94.5%	83.2%	**98.9%**	98.2%	94.7%
NX Test set 2	79.8%	76.8%	79%	65.2%	**86.6%**	77.4%	77.2%
TX2 Test set 1	98.9%	**99.7%**	97.5%	90.8%	99.5%	97.9%	97.1%
TX2 Test set 2	**95.6%**	88.8%	60.6%	89.6%	93.4%	94.0%	82.0%

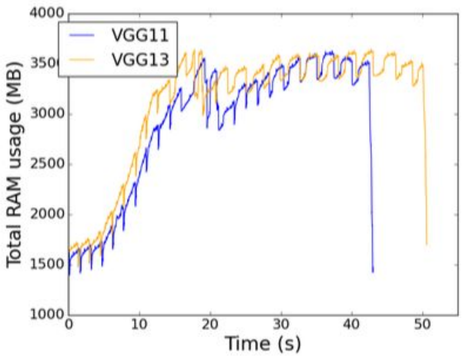
Robustness to Background Noise

- AES Encryption & Decryption running as parallel application with varied input sizes

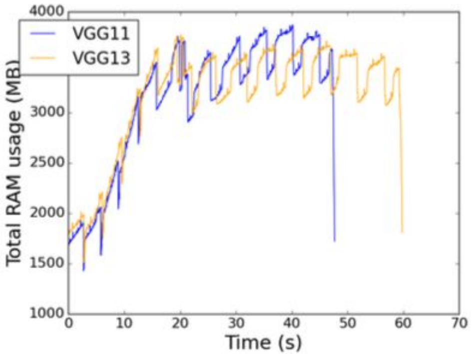
Background app \ Dataset	Test set 1	Test set 2
AES BG 10MB	86.4%	69.6%
AES BG 50MB	42.6%	38.6%
AES BG 100MB	16.9%	21.4%



(a) VGGs + AES (10MB)



(c) VGGs + AES (50MB)



(e) VGGs + AES (100MB)

Robustness to Modified Models

- Robustness to Modified Models
  - Transfer Learning on CIFAR10 – FC layer adapted, retrained
  - Input 32x32 instead of 224x224
  - Classification Accuracy: ***71.7%*** (Test Set 1), ***82.4%*** (Test Set 2)

- Robustness to Different Framework (TF)
  - Experiments repeated with Jetson Nano setup, using TensorFlow instead of Pytorch
  - Pretrained models obtained from Keras applications
  - Classification Accuracy: ***99.1%*** (Test Set 1), ***94%*** (Test Set 2)

Takeaways

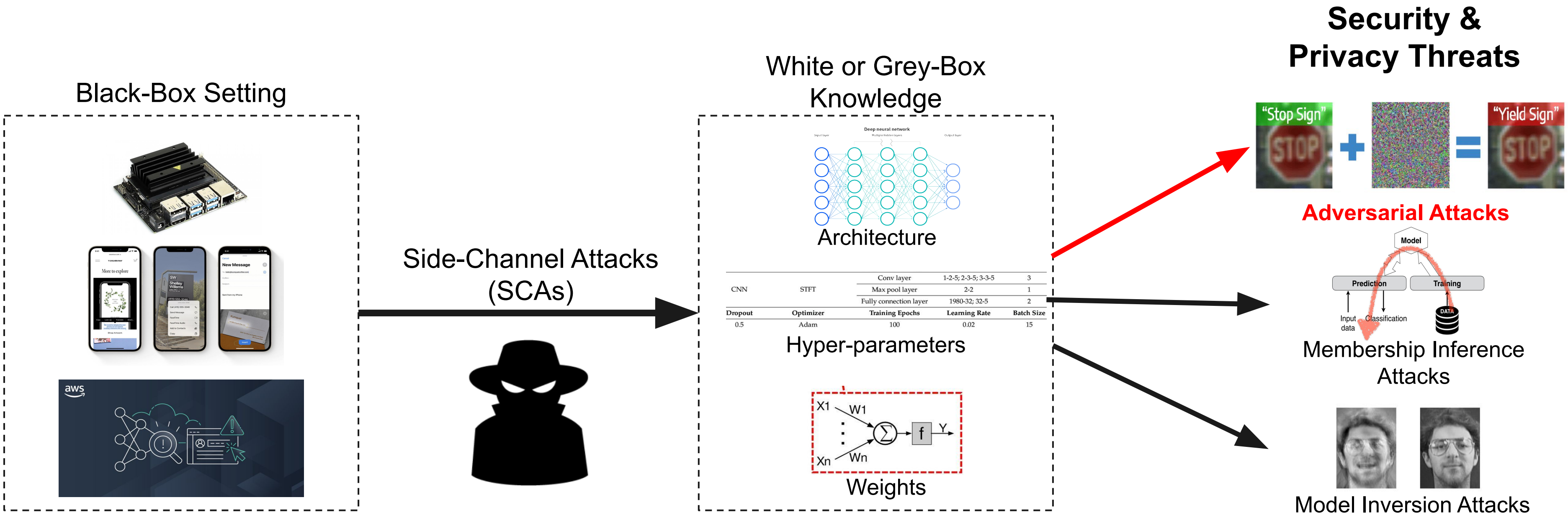
- Global-aggregate statistics (available at user-level) can leak distinguishable traces among DNN model architecture families
    - While being passive, remote, and stealthy!

- Our explored vulnerability is robust to noise, modifications to DNNs, and platform portable

- Knowledge of the extracted DNN model architecture family can improve effectiveness of ensemble adversarial attacks

Thank you! Questions?

Kartik Patwari
kpatwari@ucdavis.edu



Black-Box Setting
aws
Side-Channel Attacks
(SCAs)
White or Grey-Box
Knowledge
Deep neural network
Architecture
CNN
STFT
Conv layer
1-2-5; 2-3-5; 3-3-5
3
Max pool layer
2-2
1
Fully connection layer
1980-32; 32-5
2
Dropout
Optimizer
Training Epochs
Learning Rate
Batch Size
0.5
Adam
100
0.02
15
Hyper-parameters
X1
W1
Xn
Wn
f
Y
Weights
Security &
Privacy Threats
"Stop Sign"
"Yield Sign"
Adversarial Attacks
Model
Prediction
Training
Input data
Classification
DATA
Membership Inference
Attacks
Model Inversion Attacks