Compilation as a Defense: Enhancing DL Model Attack Robustness via Tensor Optimization

Stefan Trawicki
Lancaster University

William Hackett
Lancaster University

Lewis Birch
Lancaster University

Neeraj Suri
Lancaster University

Peter Garraghan
Lancaster University, Mindgard

Abstract

Adversarial Machine Learning (AML) is a rapidly growing field of security research, with an often overlooked area being model attacks through side-channels. Previous works show such attacks to be serious threats, though little progress has been made on efficient remediation strategies that avoid costly model re-engineering. This work demonstrates a new defense against AML side-channel attacks using model compilation techniques, namely tensor optimization. We show relative model attack effectiveness decreases of up to 43% using tensor optimization, discuss the implications, and direction of future work.

1 Introduction

1.1 Adversarial Machine Learning

Adversarial Machine Learning (AML) investigates attacks on Deep Learning (DL) models and their underlying hardware/software assets [6, 9, 13, 18]. Of note are Side-Channel Accelerator (SCA) attacks which monitor kernel metrics during DL operation to extract leaky information [9, 18]. SCA attacks can be (1) model and dataset agnostic (black box), (2) conducted using few model inferences (< 1 second), and (3) notoriously complex to protect against. Current work remediates such attacks via costly model architecture and framework modifications [9, 17].

1.2 Proposed Approach

This work proposes a new defense against SCA attacks by utilizing generalized DL compilation optimization techniques to obfuscate model architecture, and increase model robustness to attacks. Popular DL frameworks such as PyTorch [14] share common DL operator implementation libraries (e.g., CUDNN [1]). SCA attacks are designed to recognize model memory access patterns within these libraries, thus decreasing model architecture confidentiality. Our idea uses the TVM [4] compiler to apply increasing levels of machine-generated tensor optimizations (TO) to obfuscate the model architecture without manual re-engineering. We demonstrate our defense approach effectiveness against a model SCA targeting kernel L2 cache on GPUs [9].

AutoTVM is a simulated annealing technique for generating tensor optimization (TO) schedules, using a tuner to run \( n \) candidate batches (trials) on the target hardware and guide the candidate search towards optimized schedules in the search space [4]. More candidate trials typically result in faster-executing operators over time (e.g., Halide, An- sor [15,16]). We demonstrate AutoTVM using the XGB Rank cost-model [5] between 1 to 500 trials on YoloV4, RoBERTa, DenseNet121 and ResNet18 [8, 10–12]. These models range between 8m - 124m parameters, representing a diverse suite of applications (object detection, generative text, image classification). Robustness was assessed using DeepSniffer (DS) [9], an SCA attack that associates kernel L2 cache reads/writes during inference to predict model architecture, selected due to having previously shown high reconstruction success [7]. ONNX models [2] were loaded into TVM, optimized, and inference performed in the TVM runtime with kernel metrics collected by the NSYS Profiler [3]. DS was then run on the metrics to make architecture predictions with attack success measured as fidelity, the comparison of predicted and actual architecture between 0 (no similarity) and 1.0 (identical). Optimization and execution used CUDA 11.7 and the Nvidia A100 accelerator, with optimization code available on GitHub\(^1\).

2 Preliminary Findings

Initial findings demonstrate that compilation as a defense can successfully decrease the effectiveness of AML targeting the side-channel. Figure 1 shows the impact of \( n \) trials of TO on DS fidelity for each model. We observe that DS attack fidelity decreased with the number of trials conducted. After

\(^1\)https://github.com/stefanTrawicki/tdef
AML robustness, and (2) reactively applying optimization in 43%, without modifying model architecture. While maximizing optimization decreases inference time and improves robustness, it incurs large resource cost as generating and evaluating schedules is accelerator intensive (83 GPU-hours across all models/trials). In future work, we envision optimization for AML remediation via: (1) application with existing model-architecture modification approaches [17]) to improve AML robustness, and (2) reactively applying optimization in response to AML detection, making the model a ‘moving-target’. We also posit a focused technique where operators found to be conductive to SCA success are optimized first, decreasing resource requirements and attack effectiveness.

## 3 Discussion

This work investigates how tensor optimization techniques can reduce AML side-channel attack effectiveness by up to 43%, without modifying model architecture. While maximizing optimization decreases inference time and improves robustness, it incurs large resource cost as generating and evaluating schedules is accelerator intensive (83 GPU-hours across all models/trials). In future work, we envision optimization for AML remediation via: (1) application with existing model-architecture modification approaches [17]) to improve AML robustness, and (2) reactively applying optimization in response to AML detection, making the model a ‘moving-target’. We also posit a focused technique where operators found to be conductive to SCA success are optimized first, decreasing resource requirements and attack effectiveness.

### References