

FPGA-Accelerated Neural Networks for Real-Time Anomaly Detection Using High-Level Synthesis

Anila Hoxha*, Shekoufeh Neisarian*+, Talaya Farasat*, Joachim Posegga*, Elif Bilge Kavun*~

+ {shekoufeh.neisarian, elif.kavun}@barkhauseninstitut.org

* {hoxha02, neisar01, farasa01, posegga01}@ads.uni-passau.de

~ {elif_bilge.kavun}@tu-dresden.de

Motivation

- Anomaly-based network intrusion detection is a critical and challenging task
- Neural networks are applied due to their ability to recognize patterns associated with malicious activity
- Increasingly complex anomaly detection models require high-performance hardware with real-time processing capabilities
- FPGAs provide effective real-time inference acceleration, outperforming conventional hardware

Overview

- Acceleration of a Fully-Connected Neural Network (FCNN) and a one-dimensional Convolutional Neural Network (CNN), both trained on the CIDDS-001 dataset [1, 2]
- Integration of High-Level Synthesis (HLS) with hls4ml to streamline hardware design [3, 4, 5, 6]
- Evaluation and comparison of the classification and deployment performance of both models on the PYNQ-Z2 FPGA, Intel Core i7-9750H CPU, and NVIDIA GeForce RTX 2080 Ti GPU

Implementation

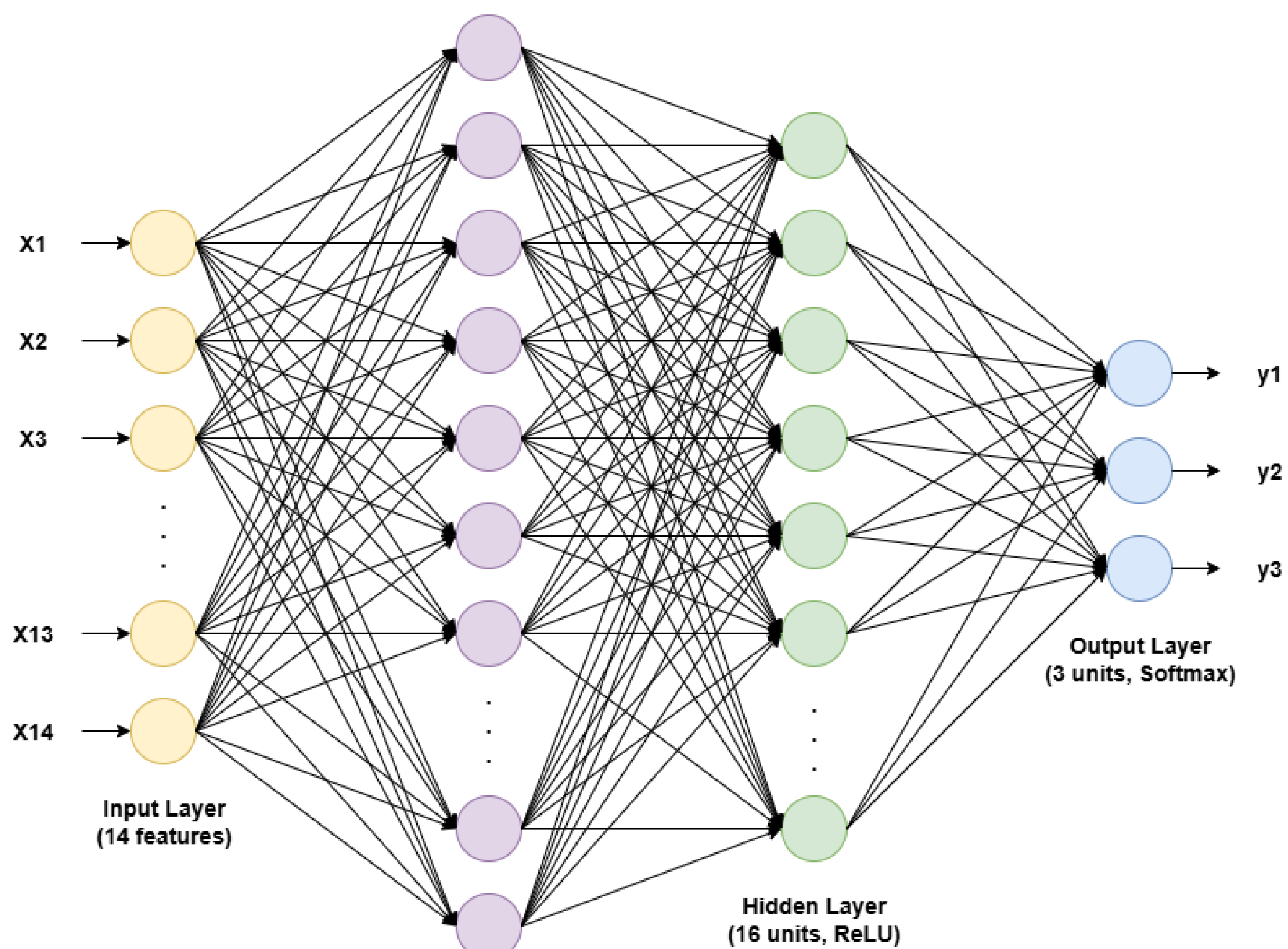


Figure 1: Proposed FCNN architecture

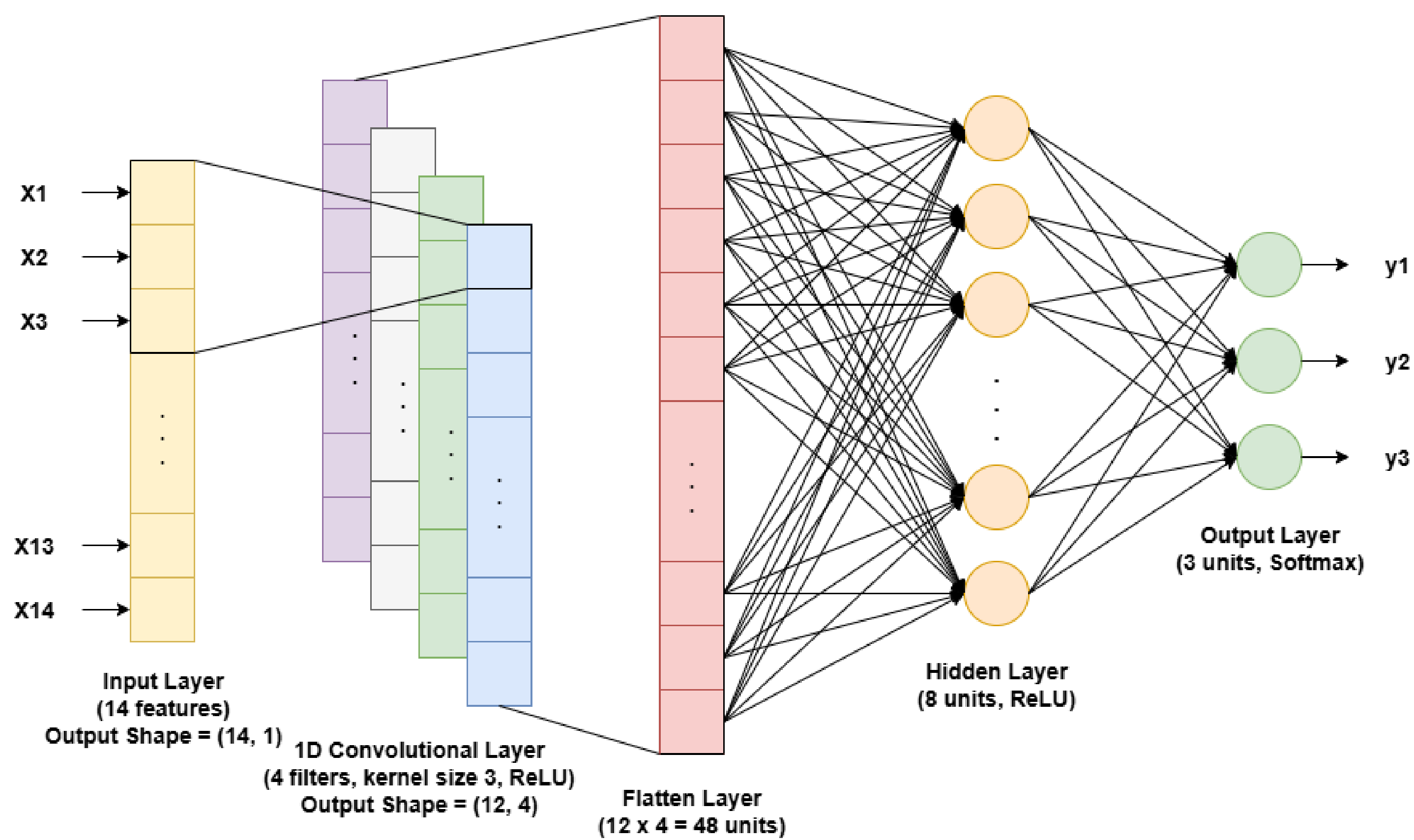


Figure 2: Proposed 1D CNN architecture

- Application of quantization-aware training with QKeras (6 bits) and pruning (80%) [7]
- Reduction of fixed-point precision for HLS configurations: FCNN <16, 6>, CNN <12, 4>
- Adjustment of reuse factor to reduce the FPGA resource utilization
- Successful synthesis of both neural networks into FPGA-firmware implementations

Results and Evaluation

Table 1: Classification accuracy (%) of both models across different hardware

Hardware	FCNN	CNN
CPU	98.16 (non-q.), 93.10 (QKeras)	98.21 (non-q.), 91.41 (QKeras)
GPU	92.88 (QKeras)	92.09 (QKeras)
FPGA	91.64 (hls4ml)	86.91 (hls4ml)

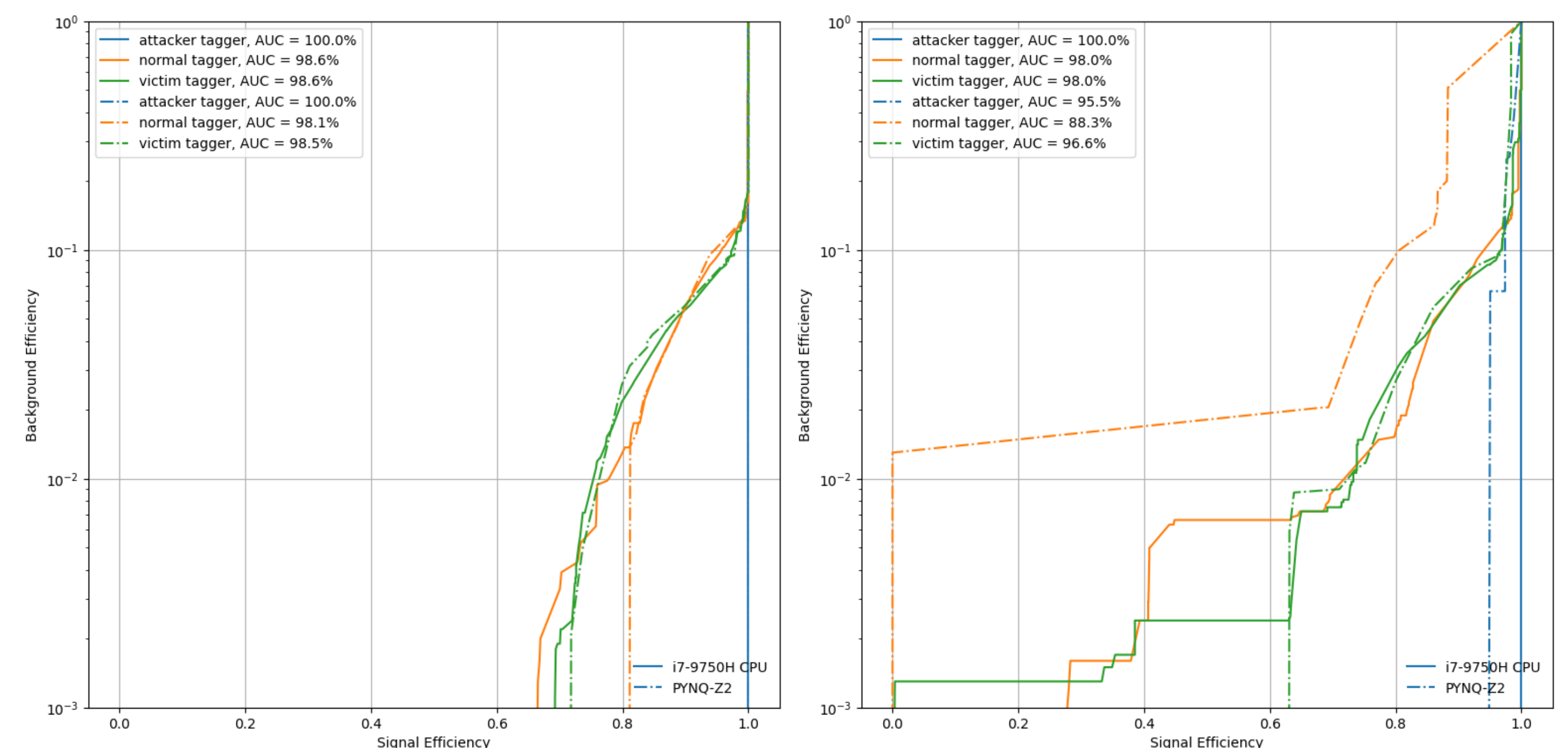


Figure 3: ROC curves for the FCNN (left) and CNN (right) models, CPU vs. GPU

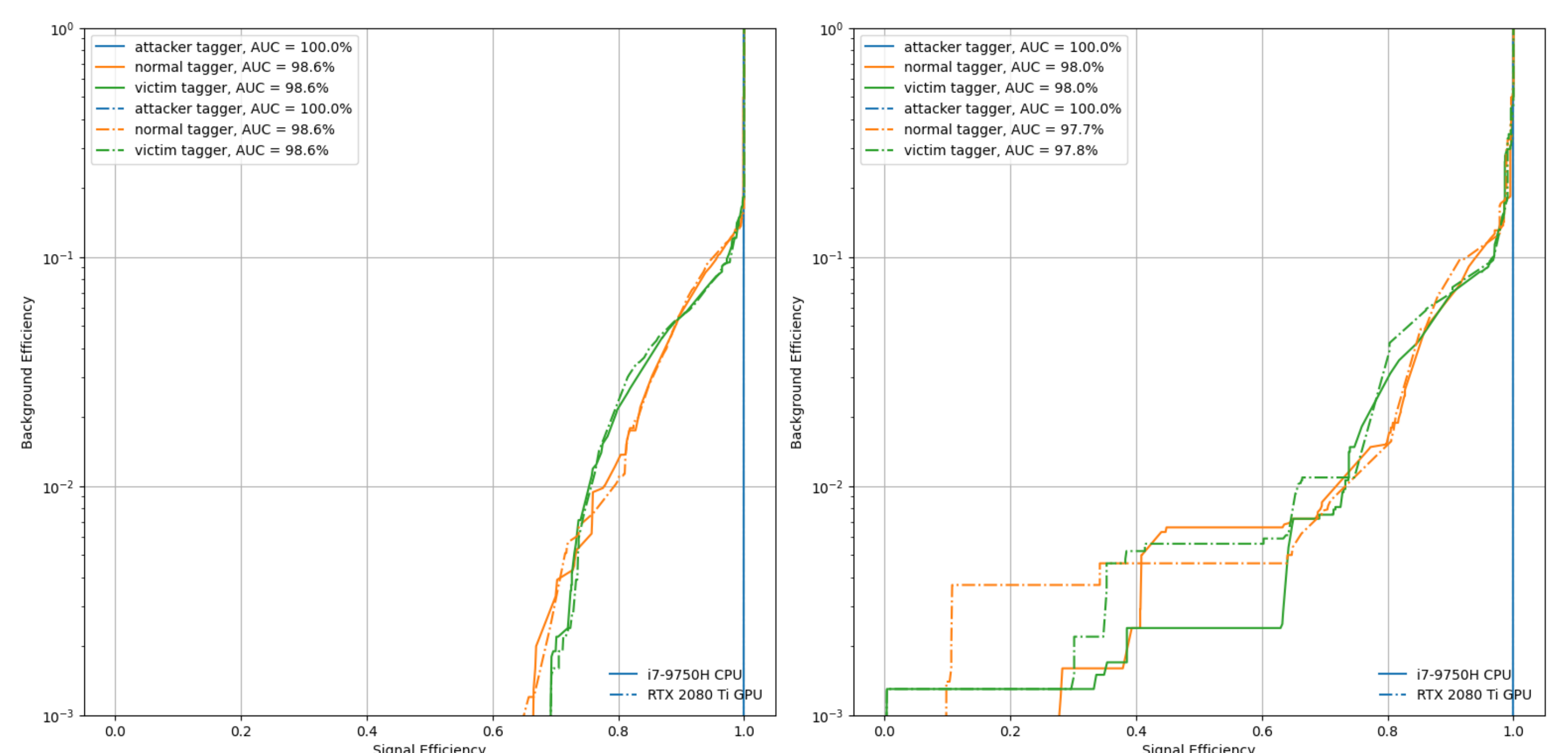


Figure 4: ROC curves for the FCNN (left) and CNN (right) models, CPU vs. FPGA

Table 2: Inference latency (s) and throughput (inferences/s)

Hardware	FCNN		CNN	
	Latency	Throughput	Latency	Throughput
CPU	3.87	3879.09	1.92	7797.37
GPU	17.97	834.90	17.22	870.95
FPGA	0.02	661404.82	0.03	566615.04

Conclusion

- **FCNN on FPGA:** ~194× faster inference & ~171× higher throughput vs. CPU | ~899× faster & ~792× higher throughput vs. GPU
- **CNN on FPGA:** 4× faster inference & ~73× higher throughput vs. CPU | 72× faster & ~651× higher throughput vs. GPU