SNN-IDS: Deploying SNN-Equivalent Intrusion Detection on a Commodity MCU NPU

Hsiu-Chi Tsai *

Posted Date: 11 March 2026

doi: 10.20944/preprints202603.0817.v2

Keywords: spiking neural network; intrusion detection system; INT8 quantization; neural processing unit; edge AI; ANN-SNN conversion; STM32N6; NSL-KDD; UNSW-NB15; post-training quantization

 doi:10.20944/preprints202603.0817.v2

Article

SNN-IDS: Deploying SNN-Equivalent Intrusion Detection on a Commodity MCU NPU

Hsiu-Chi Tsai

National Yang Ming Chiao Tung University, Hsinchu, Taiwan; hctsai1006@cs.nctu.edu.tw

Abstract

We deploy a spiking neural network (SNN)-equivalent intrusion detection system (IDS) on the STM32N6570-DK, a commodity ARM Cortex-M55 MCU with the Neural-ART NPU. Exploiting the approximate equivalence between single-timestep (T=1) SNN inference and INT8 quantized ANN inference, we compile a lightweight MLP classifier to the NPU without neuromorphic hardware. Evaluated on NSL-KDD (5-class) and UNSW-NB15 (10-class) with 10 random seeds, the ReLU model achieves 78.86±1.32% and 64.75±0.61% overall accuracy, respectively. INT8 accuracy stays within 1 percentage point of FP32 across all 24 tested calibration configurations, and layer-wise analysis shows 99.0% final prediction agreement between FP32 and INT8 models. On the NPU, the INT8 model infers in **0.46 ms** on NSL-KDD and **0.29 ms** on UNSW-NB15 (100% NPU execution), 2.7–4.2× faster than the same model on the Cortex-M55 CPU, while occupying 120.6–137.7 KB Flash and 0.5–1.25 KB RAM. Tree-based baselines (Random Forest, XGBoost) achieve higher overall accuracy on UNSW-NB15 but cannot be compiled for the NPU at all. To our knowledge, this is the first publicly documented IDS deployment on an ARM Cortex-M NPU and the first publicly documented empirical validation of T=1 SNN–ANN equivalence on commercial NPU silicon.

Keywords: spiking neural network; intrusion detection system; INT8 quantization; neural processing unit; edge AI; ANN-SNN conversion; STM32N6; NSL-KDD; UNSW-NB15; post-training quantization

1. Introduction

Deep learning has transformed network intrusion detection, but deploying these models at the network edge remains difficult. Edge IDS must operate under tight power and latency budgets, and the dominant approach today still relies on cloud offloading or high-end GPUs, neither of which fits IoT or telecom CPE (customer premises equipment) scenarios where per-device cost must stay below $10.

Spiking neural networks offer a different compute paradigm. Because SNNs process information as discrete spikes rather than dense floating-point activations, they can be extremely energy-efficient when run on neuromorphic hardware [1]. The catch is that neuromorphic chips are not commodity parts. Intel Loihi [2] is a research platform. BrainChip Akida is available commercially but carries a price premium. Neither is a standard MCU that an OEM can drop into a low-cost IoT gateway.

A line of theoretical work changes this picture. Bu et al. [3] introduced QCFS, showing that ANN activations can be mapped to discrete spike counts with minimal loss. Jiang et al. [4] provided the first T=1 ANN–SNN conversion framework via SlipReLU. More recently, Bu et al. [5] showed that when an SNN runs for exactly one timestep (T=1) with zero initial membrane potential, the threshold-and-fire operation reduces to a ReLU-like clamp, making the forward pass approximately equivalent to an INT8 quantized ANN. The practical implication: any NPU that can execute INT8 matrix multiply and ReLU can run an SNN-equivalent model without neuromorphic hardware.

We put this theory to the test. Using the STM32N6570-DK development board (ARM Cortex-M55 at 800 MHz, Neural-ART NPU rated at 600 GOPS INT8), we train an MLP for intrusion detection on two standard benchmarks (NSL-KDD and UNSW-NB15), quantize it to INT8, and deploy it on

 doi:10.20944/preprints202603.0817.v2

the NPU through ST Edge AI Developer Cloud. The classifier infers in 0.29–0.46 ms (2.7–4.2× faster than CPU-only execution) with a memory footprint under 140 KB. We focus on inference latency and memory footprint of the classifier stage on pre-extracted flow-level features; power estimates derived from ST's published application note are noted in Section 5.5 but direct on-board measurement is deferred. A complete end-to-end IDS pipeline (packet capture, flow aggregation, feature extraction) is left to future work.

Our contributions:

- We present, to our knowledge, the first publicly documented IDS classifier deployment on an ARM Cortex-M NPU (Neural-ART). Prior MCU-class IDS work used the MAX78000 [6], an AI-specialized MCU with a fixed CNN accelerator; our target is a general-purpose MCU with an attached programmable NPU.
- We provide the first publicly documented empirical validation of the T=1 SNN–ANN equivalence on commercial NPU silicon, with 99% final prediction agreement between FP32 and INT8 models.
- We conduct a quantization ablation (3 calibration methods × 2 granularities × 4 sample sizes) showing that INT8 accuracy is insensitive to calibration choices (all deviations $<$ 1 pp).
- We document that the `Floor` operator required by QCFS [3] falls back to CPU on the Neural-ART NPU, identifying a concrete barrier for deploying QCFS-based SNN models on commodity NPUs.

2. Related Work

2.1. SNN-Based Intrusion Detection

Table 1 summarizes prior work on neural-network-based IDS with SNN components or edge hardware deployment. Research on applying SNNs to network intrusion detection has grown rapidly since 2024, though nearly all published results remain in simulation. Wang et al. [7] proposed a convolutional SNN for IDS on NSL-KDD and UNSW-NB15 in software simulation. Prajwalasimha et al. [8] described an event-driven SNN-IDS and tested it in simulation. Mustafa et al. [9] combined recurrent and spiking layers for anomaly detection. Karthik et al. [10] proposed a transformer-spiking hybrid; all run only on GPU.

Two prior works deploy IDS on physical hardware. Zahm et al. [11] used the BrainChip Akida AKD1000 neuromorphic processor, reporting 98.4% accuracy at approximately 1 W. Akida is a purpose-built neuromorphic chip ($499 dev kit) with native spike processing, not a general-purpose MCU. Ngo et al. [6] deployed a lightweight MLP (1,360 parameters, 11 flow features) on the Analog Devices MAX78000, an AI-specialized MCU with a fixed CNN accelerator. They achieved 98.57% on UNSW-NB15 (binary classification: normal vs. attack) at 18 mW. The MAX78000's accelerator is hard-wired for convolutional workloads; the STM32N6 Neural-ART NPU is a programmable accelerator that supports arbitrary operator graphs within its INT8 operator set. Our work differs from both in targeting a general-purpose ARM Cortex-M MCU with an attached NPU, using multi-class classification (5-class and 10-class), and validating T=1 SNN equivalence.

Table 1. Comparison of SNN/neural-network-based IDS approaches. "On-chip" indicates deployment on physical hardware. "Task" shows the classification formulation (binary vs. multi-class). Accuracy figures are not directly comparable across different task formulations.

Paper	Year	Dataset	Task	Model	Hardware	HW Type	On-chip	Timing	Power	Acc.
Wang [7]	2024	NSL-KDD	multi	Conv SNN	GPU	GPU	No	No	No	94.7%
Zahm [11]	2024	UNSW-NB15	multi	SNN	Akida AKD1000	Neurom. ASIC	Yes	Yes	Yes	98.4%
Prajwal. [8]	2025	NSL-KDD	multi	Event SNN	GPU	GPU	No	No	No	95.4%
Mustafa [9]	2025	UNSW-NB15	binary	RNN-SNN	GPU	GPU	No	No	No	99.7%
Karthik [10]	2026	NSL-KDD	multi	Transf.-SNN	GPU	GPU	No	No	No	99.2%
Ngo [6]	2022	UNSW-NB15	binary	MLP	MAX78000	AI-spec. MCU	Yes	Yes	Yes	98.6%
This work	2026	NSL-KDD, UNSW	5/10-cls	MLP (T=1 SNN eq.)	STM32N6	MCU + NPU	Yes	Yes	No	78.9/64.8%

2.2. ANN-SNN Conversion

The theory of converting a trained ANN into an equivalent SNN has matured over several stages. Bu et al. [3] introduced QCFS, a quantization-aware activation function that maps ANN activations to

 doi:10.20944/preprints202603.0817.v2

discrete spike counts. Jiang et al. [4] proposed a unified optimization framework and the first T=1 conversion method (SlipReLU). Bu et al. [5] later analyzed the inference-scale complexity of SNNs, establishing channel-wise threshold optimization for T=1 conversion. NEXUS [12] recently achieved bit-exact ANN-to-SNN equivalence via modular arithmetic, scaling to LLaMA-2 70B. PASCAL [13] refined multi-timestep conversion with spike accumulation and adaptive layerwise calibration. NeuroFlex [14] showed that co-executing INT8 ANN and SNN layers on the same accelerator can cut the energy-delay product by 57–67%.

To our knowledge, this paper provides the first publicly documented empirical validation of the T=1 equivalence on commercial NPU silicon.

2.3. Edge AI on MCU NPUs

The STM32N6 is STMicroelectronics' first MCU with a dedicated neural processing unit [15]. Public deployments span computer vision and audio processing. The NPU's hardware operator set covers `Conv`, `Gemm`, `Relu`, `Add`, `Clip`, and `MaxPool` in INT8; operators outside this set fall back to the Cortex-M55 CPU [15]. Millar et al. [16] benchmarked micro-NPUs including the Neural-ART; their latency numbers provide a useful reference point. To our knowledge, no prior publicly documented work has deployed an IDS on any general-purpose MCU NPU. The closest prior art (HH-NIDS [6]) used a specialized MCU with a fixed CNN engine.

3. Background

3.1. T=1 SNN–ANN Equivalence

A Leaky Integrate-and-Fire (LIF) neuron with membrane potential V, weight matrix $\mathbf{W}$, and threshold θ operates as:

$$V[t] = \beta \cdot V[t-1] + \mathbf{W} \cdot \mathbf{x}[t] \tag{1}$$

$$S[t] = \Theta(V[t] - \theta) \tag{2}$$

where Θ is the Heaviside step function and β is the leak factor. At T=1 with zero initial state ($V[0] = 0$), the leak term $\beta \cdot V[0]$ vanishes regardless of β, so Eq. 1 simplifies to $V[1] = \mathbf{W} \cdot \mathbf{x}$.

Multiple works have established that INT8 post-training quantization produces a discretization that closely approximates the threshold-and-fire operation [3–5]. In practice, any NPU executing INT8 `Gemm` + `Relu` is performing SNN-equivalent computation, with accuracy gaps typically below 1 pp on standard benchmarks.

3.2. QCFS Activation

QCFS [3] maps ANN activations to discrete spike counts: $\mathrm{QCFS}(x) = \lfloor \mathrm{clip}(x/s,\, 0,\, L) \rfloor \cdot s$, where s=θ/L is the quantization step and L is the number of levels. Its ONNX graph requires `Floor`, which is absent from the Neural-ART operator set (Section 5.5).

3.3. STM32N6 Neural-ART NPU

The STM32N6570-DK pairs an ARM Cortex-M55 at 800 MHz with ST's Neural-ART NPU, rated at 600 GOPS for INT8 workloads [15]. The NPU natively supports `Conv`, `Gemm`, `Relu`, `Add`, `Clip`, and `MaxPool` in INT8. Operators outside this set are dispatched to the CPU, incurring cache-maintenance overhead for NPU↔CPU data transfers.

4. Method

Figure 1 shows the deployment pipeline. Both paths share the same training and export procedure; they diverge only in the activation function.

 doi:10.20944/preprints202603.0817.v2

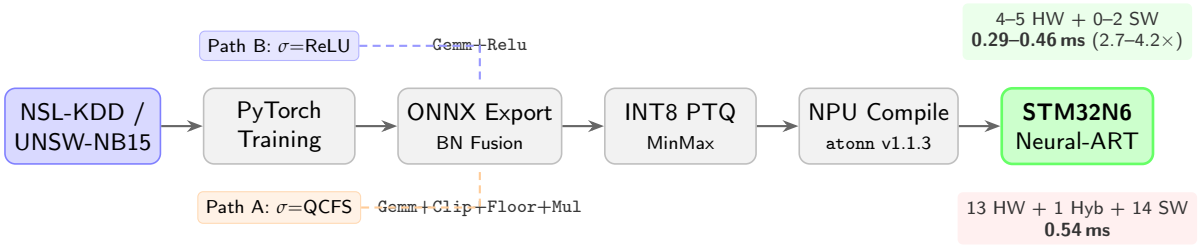

Figure 1. Deployment pipeline. Path B (ReLU) produces a Gemm+Relu graph that maps to the NPU. Path A (QCFS) adds Floor, which falls back to CPU.

4.1. Model Architecture

We use a four-layer MLP with BatchNorm:

$$\texttt{Lin}(d{\to}256) \to \texttt{BN} \to \sigma \to \texttt{Lin}(256{\to}256) \to \texttt{BN} \to \sigma$$
$$\to \texttt{Lin}(256{\to}128) \to \texttt{BN} \to \sigma \to \texttt{Lin}(128{\to}C)$$

where d is the input dimension (41 for NSL-KDD, 34 for UNSW-NB15), C is the number of classes (5 or 10), and σ is ReLU (Path B) or QCFS L=4 (Path A). The model has 111,365 parameters (NSL-KDD) or 110,218 (UNSW-NB15). At ONNX export, BatchNorm is folded into the preceding Linear layer.

4.2. Datasets

NSL-KDD [17] contains 125,973 training and 22,544 test instances with 41 features. We use 5-class grouping: *normal*, *DoS*, *Probe*, *R2L*, *U2R*. Three categorical features are label-encoded; 38 continuous features are z-score normalized using training-set statistics.

UNSW-NB15 [18] contains 175,341 training and 82,332 test instances with 34 features and 10 attack categories: *Analysis*, *Backdoor*, *DoS*, *Exploits*, *Fuzzers*, *Generic*, *Normal*, *Reconnaissance*, *Shellcode*, *Worms*. Three categorical features (`proto`, `service`, `state`) are label-encoded; remaining features are z-score normalized.

4.3. Training

Both paths use Adam (lr $= 10^{-3}$, weight decay 10^{-5}), cosine annealing over 80 epochs, batch size 512, and inverse-frequency class weighting. All experiments are repeated with 10 random seeds (0–9); we report mean $\pm$ standard deviation. Training uses PyTorch 2.10 on an NVIDIA DGX Spark (GB10 GPU).

4.4. INT8 Post-Training Quantization

FP32 ONNX models are quantized to INT8 using ONNX Runtime static quantization. The default configuration uses MinMax calibration on 1,000 randomly sampled training instances with per-tensor quantization. Section 5.4 explores the sensitivity to calibration method, granularity, and sample size.

4.5. NPU Compilation

Models are compiled for the STM32N6570-DK using ST Edge AI Developer Cloud v4.0.0 (compiler: Neural-ART `atonn` v1.1.3). The compiler partitions the ONNX graph into NPU-executable epochs (hardware or hybrid) and CPU-only epochs (software).

5. Experiments

5.1. Multi-Seed Classification Results

Table 2 reports classification results across 10 random seeds. On NSL-KDD, the ReLU MLP achieves the best overall accuracy (78.86±1.32%) and macro F1 (59.20±2.80%), outperforming both tree baselines and QCFS. A Wilcoxon signed-rank test yields p=0.037 for overall accuracy (ReLU >

 doi:10.20944/preprints202603.0817.v2

QCFS) but p=0.232 for macro F1 (not significant). This overturns a single-seed observation (seed 42) where QCFS appeared superior — a cautionary example of why multi-seed evaluation matters.

Table 2. Test accuracy (%, mean±std over 10 seeds) on NSL-KDD and UNSW-NB15. Best per dataset in **bold**.

Dataset	Model	Overall	Macro Acc	Macro F1
NSL-KDD	ReLU MLP	**78.86±1.32**	58.12±1.88	**59.20±2.80**
	QCFS L=4	77.82±1.11	57.08±1.62	57.29±2.77
	Rand. Forest	73.84±0.19	—	47.13±0.33
	XGBoost	76.86±0.00	—	55.52±0.00
UNSW-NB15	ReLU MLP	64.75±0.61	**61.30±0.76**	40.29±0.90
	Rand. Forest	**69.46±0.10**	—	**48.63±0.38**
	XGBoost	67.10±0.00	—	47.46±0.00

On UNSW-NB15 (10-class), the MLP achieves 64.75±0.61% overall accuracy, lower than Random Forest (69.46±0.10%). However, the MLP achieves the highest macro accuracy (61.30±0.76%), which reflects better balance across minority classes from inverse-frequency weighting. The gap in overall accuracy reflects the difficulty of UNSW-NB15's 10-class problem with extreme imbalance (Worms: 130 training / 44 test samples vs. Normal: 56,000 training).

5.2. Per-Class Performance

Table 3 shows per-class metrics on NSL-KDD. DoS and Normal achieve strong F1 (> 82%), but R2L and U2R remain challenging. R2L exhibits high precision (93.5%) but low recall (25.0%), indicating that the model correctly identifies R2L when it predicts the class but misses most R2L samples. U2R has only 52 training samples (0.04% of the training set); no amount of class weighting can fully compensate. Figure 2 visualizes the confusion pattern: R2L samples overwhelmingly fall into the Normal bin (64.4%), and U2R scatters across Probe (50.9%) and Normal (24.0%).

Table 3. Per-class metrics (%, mean±std) for ReLU MLP on NSL-KDD (10 seeds). P = precision, R = recall, F1 = F1-score.

Class	P	R	F1	Support
DoS	95.9±0.5	80.2±2.7	87.3±1.5	7,458
Probe	77.3±1.6	70.3±5.3	73.6±3.1	2,421
R2L	93.5±1.5	25.0±9.1	38.7±11.5	2,754
U2R	13.6±10.6	18.6±3.3	13.8±4.1	200
Normal	72.2±1.4	96.5±0.5	82.6±0.9	9,711

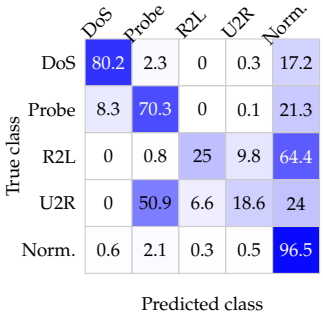

Figure 2. Normalized confusion matrix (%, 10-seed mean) for ReLU MLP on NSL-KDD. R2L→Normal (64.4%) and U2R→Probe (50.9%) are the dominant misclassification patterns.

Table 4 shows per-class results on UNSW-NB15. Generic dominates (F1=97.9%), while extreme minority classes collapse: Worms (44 test samples, F1=9.0%), Analysis (F1=2.6%). Normal achieves

 doi:10.20944/preprints202603.0817.v2

98.8% precision but only 57.6% recall — inverse-frequency weighting shifts predictions toward attack categories at the expense of the majority class.

Table 4. Per-class metrics (%, mean±std) for ReLU MLP on UNSW-NB15 (10 seeds). Classes sorted by F1 descending.

Class	P	R	F1	Support
Generic	99.8±0.1	96.1±0.4	97.9±0.2	18,871
Reconnaissance	68.4±4.9	83.6±1.1	75.1±2.7	3,496
Normal	98.8±0.3	57.6±0.6	72.7±0.4	37,000
Exploits	78.8±0.9	49.7±1.4	60.9±0.9	11,132
Fuzzers	27.2±0.5	65.4±1.6	38.4±0.6	6,062
DoS	29.4±3.5	15.9±8.8	19.5±6.6	4,089
Shellcode	8.9±1.1	86.0±3.6	16.1±1.7	378
Backdoor	5.8±0.5	66.2±15.9	10.5±1.0	583
Worms	4.8±0.9	83.6±5.1	9.0±1.5	44
Analysis	1.5±1.3	9.0±9.4	2.6±2.2	677

5.3. Layer-Wise FP32 vs INT8 Equivalence

Table 5 presents layer-wise activation comparison between the FP32 and INT8 models. Individual hidden layers show moderate cosine similarity (0.65–0.68) due to per-neuron quantization noise, but the logit layer recovers to 0.978. The overall prediction agreement is **99.0%** (per-class: Normal 99.8%, DoS 99.4%, R2L 99.3%, Probe 95.4%, U2R 85.7%). INT8 quantization introduces bounded perturbation at each layer, but the accumulated effect on the final classification decision is negligible. From the $T{=}1$ SNN perspective, this means the quantized SNN (INT8 model) closely reproduces the firing decisions of the original SNN (FP32 ReLU model with $V[0]{=}0$).

Table 5. Layer-wise comparison between FP32 and INT8 ONNX models on 1,000 NSL-KDD test samples. Cosine similarity is computed on flattened activation vectors.

Layer	Shape	Cosine Sim	MAE	L_∞
Relu_0 (256)	1000×256	0.667	0.435	43.7
Relu_1 (256)	1000×256	0.655	0.438	62.6
Relu_2 (128)	1000×128	0.683	0.400	65.2
Logits (5)	1000×5	**0.978**	0.103	19.8

5.4. Quantization Ablation

Table 6 shows a representative subset of the full 24-configuration ablation (3 methods × 2 granularities × 4 sample sizes; full results in the supplementary material). Across all configurations, accuracy deviates from FP32 by at most 0.82 pp, and several INT8 configurations actually *improve* over FP32 (likely due to regularization effects of quantization noise). This stability is expected for a small model (111K parameters, 4 layers): the quantization grid resolution (256 INT8 levels) is sufficient to represent the narrow activation distributions of a shallow MLP. The practical implication is that $T{=}1$ equivalence is preserved regardless of calibration choices.

 doi:10.20944/preprints202603.0817.v2

Table 6. INT8 quantization ablation on NSL-KDD. FP32 baseline: 76.45% (seed 42, used for ONNX export). Δ = FP32 − INT8 (positive = degradation). All 24 configurations show $|\Delta| < 1$ pp.

Method	Granularity	Cal. Samples	INT8 %	Δ
MinMax	per-tensor	100	76.59	−0.14
MinMax	per-tensor	1000	76.38	+0.08
MinMax	per-tensor	5000	77.27	−0.82
MinMax	per-channel	1000	76.51	−0.05
Entropy	per-tensor	1000	76.38	+0.08
Entropy	per-channel	1000	76.51	−0.05
Percentile	per-tensor	1000	76.39	+0.07
Percentile	per-channel	1000	76.28	+0.17

5.5. NPU Hardware Benchmark

Table 7 presents measurements from the STM32N6570-DK. We first establish a CPU baseline: the same ReLU MLP compiled as FP32 runs entirely on the Cortex-M55 CPU (0 HW epochs, 11 SW epochs) at 1.24 ms (NSL-KDD) and 1.23 ms (UNSW-NB15).

Table 7. Hardware benchmark on STM32N6570-DK (Cortex-M55 @ 800 MHz + Neural-ART NPU). HW = NPU-only epochs, Hyb = mixed, SW = CPU-only.

Model	Dataset	Time (ms)	HW	Hyb	SW	Flash (KB)	RAM (KB)
ReLU FP32	NSL-KDD	1.24	0	0	11	466.4	2.17
ReLU INT8	NSL-KDD	0.46	5	1	2	137.7	1.25
ReLU FP32	UNSW-NB15	1.23	0	0	11	461.9	2.14
ReLU INT8	UNSW-NB15	**0.29**	4	0	0	120.6	0.50
QCFS INT8	NSL-KDD	0.54	13	1	14	138.0	2.00

INT8 quantization combined with NPU acceleration yields **2.7**× speedup on NSL-KDD (1.24→0.46 ms) and **4.2**× on UNSW-NB15 (1.23→0.29 ms). The larger speedup on UNSW-NB15 reflects 100% NPU execution (4 HW epochs, zero SW fallback), while NSL-KDD retains 2 SW epochs. Flash memory shrinks 3.4–3.8× from FP32 to INT8. ST Application Note AN5946 reports ∼150 mW average power during NPU inference at nominal frequency [19]. Combining this with our measured latency yields an estimated ∼69 µJ/inference (NSL-KDD) and ∼44 µJ/inference (UNSW-NB15). These figures are derived from the reference workload in AN5946, not direct measurement of our model, and should be validated with on-board instrumentation.

QCFS INT8 runs `Gemm` on the NPU but each `Floor` triggers a CPU fallback via `DequantizeLinear` → `Floor(float)` → `QuantizeLinear`, adding 0.08 ms (+17.4%) over ReLU INT8. The `Floor` operator is absent from ST's published NPU operator list [15].

5.6. Tree-Based Baseline Comparison

Random Forest (100 trees, max depth 20) and XGBoost (100 trees, max depth 6) serve as non-neural baselines. Both use balanced class weighting and are evaluated across 10 seeds. ONNX export succeeds for Random Forest (using `skl2onnx`), producing models with `TreeEnsembleClassifier` operators. These operators are not in the Neural-ART NPU's supported set and run entirely on the Cortex-M55 CPU.

On NSL-KDD, the MLP outperforms both tree models in overall accuracy and macro F1 (Table 2), while also being the only model eligible for NPU acceleration. On UNSW-NB15, Random Forest achieves higher overall accuracy (69.46% vs. 64.75%), but the MLP achieves higher macro accuracy (61.30%), suggesting better minority-class handling. The tree models cannot run on the STM32N6 at all: ST Edge AI Core rejects the `TreeEnsembleClassifier` operator with "NOT IMPLEMENTED." The

 doi:10.20944/preprints202603.0817.v2

MLP is therefore the only architecture eligible for NPU acceleration, achieving 0.29 ms inference on hardware where tree models have no path to execution.

6. Discussion

6.1. Validating T=1 Equivalence in Practice

The layer-wise analysis (Section 5.3) provides direct evidence that T=1 SNN–ANN equivalence survives INT8 quantization. Individual hidden layers show cosine similarity of 0.65–0.68 between FP32 and INT8 activations. This is expected: INT8 maps each activation to one of 256 discrete levels, introducing bounded per-neuron noise. However, the noise does not flip classification decisions — 99% of test samples receive the same predicted class from FP32 and INT8 models. The quantization ablation (Section 5.4) supports this conclusion: all 24 calibration configurations fall within 0.82 pp of FP32, ruling out the possibility that the equivalence depends on a particular calibration choice.

6.2. Why NPU When Trees Are More Accurate?

On UNSW-NB15, Random Forest achieves 69.46% overall accuracy vs. the MLP's 64.75%. A natural question is whether the NPU adds value. Three factors favor NPU deployment: (1) `TreeEnsembleClassifier` is not implemented in ST Edge AI Core, so tree models have no execution path on the STM32N6; (2) NPU inference frees the Cortex-M55 CPU for concurrent RTOS tasks (packet capture, feature extraction), whereas CPU-only inference blocks the processor for 1.2 ms per sample; (3) INT8 NPU execution reduces Flash by 3.4–3.8× over FP32, which matters on memory-constrained MCUs. The MLP is not the most accurate classifier on every dataset, but it is the only architecture deployable on the NPU hardware. More broadly, this work targets a different design point from GPU-based SNN-IDS research (Table 1): sub-millisecond inference on a commodity MCU rather than state-of-the-art accuracy on unconstrained hardware. The gap in absolute accuracy reflects multi-class evaluation (5/10-class vs. binary), multi-seed averaging (mean vs. best-of-one), and NPU operator constraints (shallow MLP vs. deep architectures).

6.3. QCFS and Floor Fallback

With multi-seed evaluation, ReLU outperforms QCFS on NSL-KDD (p=0.037), overturning a single-seed artifact. Combined with 17.4% latency overhead from `Floor` CPU fallback, QCFS offers no advantage on this NPU target.

6.4. Comparison with HH-NIDS

Ngo et al. [6] achieved 98.57% on UNSW-NB15 with the MAX78000, but used binary classification (normal vs. attack) with 11 flow features. Our 10-class multi-class problem is substantially harder. The MAX78000 is an AI-specialized MCU with a fixed CNN accelerator (442 KB weight memory); the STM32N6 is a general-purpose MCU where the NPU is one peripheral among many (GPIO, UART, Ethernet, display controller). The two represent different deployment philosophies: the MAX78000 is designed specifically for always-on AI workloads, while the STM32N6 is a general-purpose platform that can accommodate AI inference as one of many concurrent tasks under an RTOS.

6.5. Limitations

Five limitations bound our results. First, NSL-KDD is a legacy benchmark (1998 DARPA traffic) and UNSW-NB15, while more modern, still does not represent current enterprise traffic. Second, we measure inference latency on the NPU but not power consumption. Third, the evaluation covers classifier inference only, not the complete IDS pipeline (packet capture, flow aggregation, feature extraction). Fourth, the MLP architecture is shallow; deeper models might improve minority-class performance but require verifying NPU operator compatibility. Fifth, UNSW-NB15 accuracy (64.75%) is below tree baselines (69.46%); the 10-class problem with extreme imbalance (Worms: 44 test samples) is inherently difficult, and the MLP may benefit from feature selection or architecture search on this dataset.

 doi:10.20944/preprints202603.0817.v2

6.6. Future Directions

RTOS integration. The 0.29–0.46 ms inference time fits within a 1 ms RTOS tick. Integration with µT-Kernel 3.0 on the STM32N6, with lwIP for packet processing and the NPU for classification, is the natural next step toward a complete edge IDS.

Modern datasets. Retraining on CICIDS2017 or CSE-CIC-IDS2018 would test generalization.

Power measurement. Measuring Joules per inference on the NPU path would enable direct comparison with HH-NIDS's published 18 mW figure.

Other MCU NPUs. Testing on NXP i.MX RT1180 (eIQ Neutron), Renesas RA8 (Ethos-U55), and Alif Ensemble would establish cross-platform generality.

7. Conclusion

We trained an MLP for intrusion detection on NSL-KDD (5-class) and UNSW-NB15 (10-class), quantized it to INT8, and deployed it on the STM32N6570-DK Neural-ART NPU. Multi-seed evaluation (10 seeds) shows 78.86±1.32% overall accuracy on NSL-KDD and 64.75±0.61% on UNSW-NB15. NPU inference runs 2.7–4.2× faster than CPU-only execution (0.29–0.46 ms vs. 1.2 ms), with 3.4–3.8× Flash reduction. A 24-configuration quantization ablation shows INT8 accuracy varies by less than 1 pp across calibration choices. Layer-wise analysis shows 99% final prediction agreement between FP32 and INT8 models — evidence that T=1 SNN–ANN equivalence holds on commercial NPU silicon.

The QCFS experiment adds nuance: the `Floor` operator has no NPU support, forcing CPU fallback with 17.4% latency overhead. For commodity MCU NPUs with INT8 operator sets, ReLU remains the pragmatic activation for SNN-equivalent deployment.

All code, trained models, and benchmark results are publicly available at https://github.com/thc1006/SpikeIDS-MCU.

References

1. Roy, K.; Jaiswal, A.; Panda, P. Towards Spike-Based Machine Intelligence with Neuromorphic Computing. *Nature* **2019**, *575*, 607–617. https://doi.org/10.1038/s41586-019-1677-2.
2. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* **2018**, *38*, 82–99. https://doi.org/10.1109/MM.2018.112130359.
3. Bu, T.; Fang, W.; Ding, J.; Dai, P.; Yu, Z.; Huang, T. Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks. In Proceedings of the International Conference on Learning Representations (ICLR), 2022.
4. Jiang, H.; Anumasa, S.; De Masi, G.; Xiong, H.; Gu, B. A Unified Optimization Framework of ANN-SNN Conversion: Towards Optimal Mapping from Activation Values to Firing Rates. In Proceedings of the International Conference on Machine Learning (ICML), 2023, Vol. 202, *Proceedings of Machine Learning Research*.
5. Bu, T.; Li, M.; Yu, Z. Inference-Scale Complexity in ANN-SNN Conversion for High-Performance and Low-Power Applications. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025. arXiv:2409.03368.
6. Ngo, D.M.; Lightbody, D.; Temko, A.; Murphy, C.C.; Popovici, E. HH-NIDS: Hardware Heterogeneous Network Intrusion Detection System Using MAX78000. *Future Internet* **2022**, *15*, 9. https://doi.org/10.3390/fi15010009.
7. Wang, Z.; Ghaleb, F.A.; Zainal, A.; Siraj, M.M.; Lu, X. An efficient intrusion detection model based on convolutional spiking neural network. *Scientific Reports* **2024**, *14*, 7054. https://doi.org/10.1038/s41598-024-57691-x.
8. Prajwalasimha, S.N.; Sumathi, D.; Shelke, N.; Pimpalkar, A.; Saini, D.K.J.B.; Kumar, G.H. Event-Driven Intrusion Detection Systems using Spiking Neural Networks for Edge and IoT Security. In Proceedings of the 5th International Conference on Soft Computing for Security Applications (ICSCSA). IEEE, 2025, pp. 41–47. https://doi.org/10.1109/ICSCSA66339.2025.11171294.

 doi:10.20944/preprints202603.0817.v2

9. Mustafa, M.; Babiker, S.M.E.; Mustafa, Y.E.A. Hybrid recurrent with spiking neural network model for enhanced anomaly prediction in IoT networks security. *Frontiers in Artificial Intelligence* **2025**, *8*. https://doi.org/10.3389/frai.2025.1651516.
10. Karthik, M.G.; Keerthika, V.; Mantena, S.V.; Siri, D.; Yeluri, L.P.; Lella, K.K.; Ganesh, B.R. Energy-efficient intrusion detection with a protocol-aware transformer–spiking hybrid model. *Scientific Reports* **2026**, *16*, 7095. https://doi.org/10.1038/s41598-026-37367-4.
11. Zahm, W.; Nishibuchi, G.; Jose, A.; Chelian, S.; Vasan, S. Low-Power Cybersecurity Attack Detection Using Deep Learning on Neuromorphic Technologies. Technical report, CSIAC, 2024.
12. NEXUS Team. NEXUS: Bit-Exact ANN-to-SNN Equivalence via Modular Arithmetic. *arXiv preprint arXiv:2601.21279* **2026**.
13. Ramesh, P.; Srinivasan, G. PASCAL: Precise and Efficient ANN-SNN Conversion using Spike Accumulation and Adaptive Layerwise Activation. *Transactions on Machine Learning Research* **2025**.
14. Manjunath, V.; Ramesh, P.; Srinivasan, G. NeuroFlex: Column-Exact ANN-SNN Co-Execution Accelerator with Cost-Guided Scheduling. *arXiv preprint arXiv:2511.05215* **2025**.
15. STMicroelectronics. *ST Neural-ART NPU Concepts*, 2025.
16. Millar, I.; et al. Benchmarking Micro Neural Processing Units. *arXiv preprint arXiv:2503.22567* **2025**.
17. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528.
18. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. In Proceedings of the Military Communications and Information Systems Conference (MilCIS). IEEE, 2015, pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942.
19. STMicroelectronics. *How to Optimize Low-Power Modes on STM32N6 MCUs*, 2025. Application Note AN5946.