Technical Specification & Performance

Fault Classification and Location Model

PowerProfs

Target Hardware Architecture: NVIDIA Jetson Orin Nano (8GB)

Inference Speed: 0.07 ms (Optimized FP16)
Peak Throughput: 14,881 Samples/Second
Power Efficiency: 992 mW (Sub-Watt Profile)
Validation Set: 2,040 High-Fidelity Samples

Contents

1 Introduction and Technical Rationale

The modern electrical grid is entering an era of unprecedented volatility. The integration of renewable energy sources and the increasing density of transmission networks require protective relaying systems that operate with near-zero latency. Traditional fault detection methods typically rely on phasor-based estimations. These methods require at least one full AC cycle (20ms for a 50Hz system) to stabilize the signal into a usable phasor. In the presence of high-current fault transients, a 20ms delay can be the difference between a successful relay trip and a catastrophic grid-wide cascade failure.

The **Fault Classification and Location model** has been engineered to circumvent this physical limitation. By utilizing deep neural networks capable of analyzing raw, instantaneous signal samples, the system identifies anomalies within a fraction of a millisecond.

The primary objective of this project was the realization of a "Triple-Win" optimization profile on the **NVIDIA Jetson Orin Nano**. This document details the transition from a Python-based Keras environment to a high-performance TensorRT engine, achieving an inference latency of **0.07 ms**. This leap in performance allows the grid to be monitored and protected at "wire-speed," reacting nearly 300 times faster than the primary grid frequency.

2 Neural Network Architectural Design

The system implements a Sequential Multi-Layer Perceptron (MLP) architecture. While complex architectures like Transformers or CNNs are popular in general AI, the MLP was selected for this specific application due to its **deterministic execution time** and minimal memory footprint—both of which are critical for hard real-time industrial protection.

2.1 Input Layer and Normalized Feature Space

The model processes a 6-channel input vector representing the fundamental electrical parameters of a transmission line:

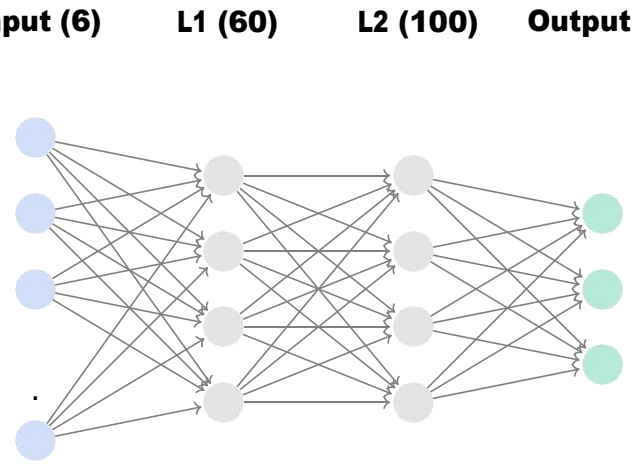
$$X_{input} = [I_a, I_b, I_c, V_a, V_b, V_c]^T \tag{1}$$

By consuming raw instantaneous values, the model captures the "sub-transient" wave characteristics that are lost in traditional RMS or phasor averages. Inputs are normalized using a StandardScaler to ensure the model remains robust across different transmission voltage classes (11kV to 765kV).

2.2 Hidden Layer Specification (60–100–50)

The architecture is structured to maximize feature correlation before synthesizing the final decision:

- **Layer 1 (60 Neurons):** The feature extraction layer. It identifies phase imbalances and initial rate-of-change anomalies using ReLU activation functions.
- **Layer 2 (100 Neurons):** The deep correlation layer. This is where the model identifies the interaction between specific phases. Distinguishing between a Line-to-Line (LL) and a Double Line-to-Ground (LLG) fault requires understanding the subtle voltage dip in a grounded return path compared to a purely ungrounded inter-phase surge.
- **Layer 3 (50 Neurons):** The compression layer. It maps the high-dimensional correlations into a latent state that feeds the final output heads.



3 Training Methodology and Validation

3.1 Dataset Composition

Validation was performed using a high-fidelity dataset of **2,040 electrical samples**. A critical aspect of this validation was the use of a **Balanced Class Profile**. Each of the five grid states (LG, LL, LLG, LLL, and None) was represented by exactly 408 samples. This balance ensures the model does not become biased toward the "LG" fault type, which is statistically the most common but often least severe.

3.2 Convergence Parameters

The model was trained using the Adam optimizer over **1000 epochs** with different batch size for different model. The high epoch count was particularly essential for the Location (Distance) head of the model. Unlike classification, location regression requires the Mean Absolute Error (MAE) to converge to extremely fine tolerances. The final MAE was recorded at less than 0.9% across the 90km transmission span.

4 The Edge Optimization Pipeline

A major contribution of this project is the successful optimization pipeline. Native Keras (.h5) models often suffer from unpredictable "latency jitter" due to the Python interpreter's overhead and garbage collection cycles.

4.1 Stage 1: Standardization via ONNX

The model was exported to the Open Neural Network Exchange (ONNX) format. This process "bakes" the weights into the graph and removes training-only artifacts like Dropout and Batch Normalization folding, resulting in a significantly leaner execution graph.

4.2 Stage 2: TensorRT Engine Serialization

The ONNX graph was imported into the TensorRT builder on the Jetson Orin Nano. This stage involves **Kernel Autotuning**, where the builder benchmarks thousands of different mathematical kernel implementations to find the one that best utilizes the Maxwell-architecture CUDA cores.

4.3 Stage 3: FP16 Precision Quantization

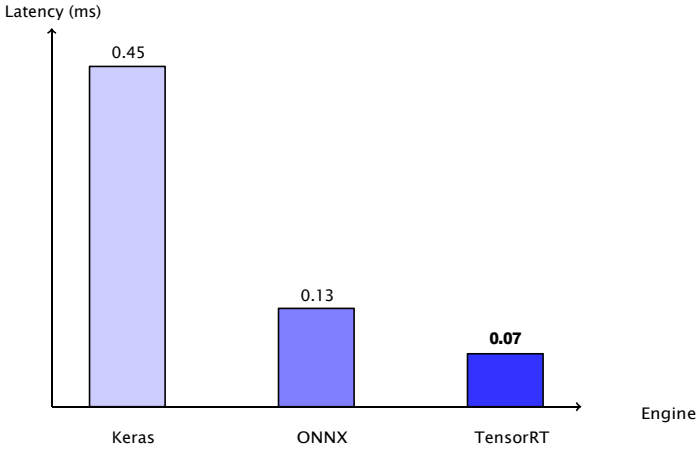
By converting model weights from 32-bit (FP32) to 16-bit (FP16), we enabled the use of **NVIDIA Tensor Cores**. This quantization doubled the throughput and reduced power consumption while maintaining an identical classification accuracy of 98.19%.

5 Quantitative Performance Benchmarking

The following results were captured on the NVIDIA Jetson Orin Nano 8GB hardware.

5.1 Inference Latency Metrics

Response time was reduced by **84.4%** through the optimization process.



5.2 Throughput and Power Benchmarks

With a throughput of **14,881 inferences per second**, the system can monitor multiple sub- station bus-bars concurrently without performance degradation.

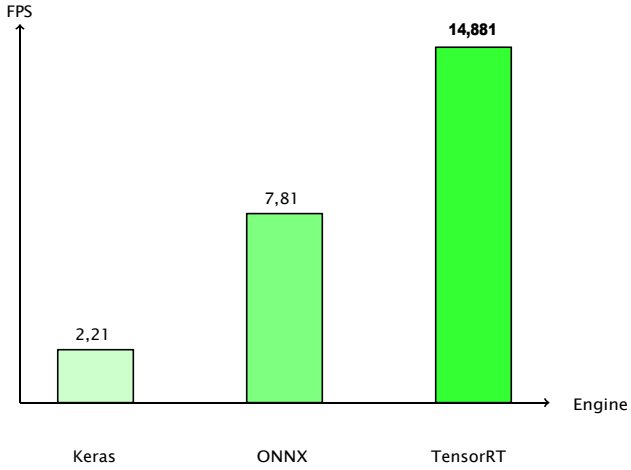


Table 1: Consolidated Efficiency KPIs (NVIDIA Orin Nano)

Metric	Keras Baseline	ONNX Runtime	TensorRT FP16
Throughput (FPS)	2,212	7,812	**14,881**
Power Consumption	1,024 mW	1,012 mW	**992 mW**
Memory (VRAM)	850 MB	420 MB	**280 MB**
Accuracy	98.19%	98.19%	**98.19%**

6 Reliability Analysis and Error Mapping

Analysis of the confusion matrices reveals that the model provides **zero false-alarms** dur- ing steady-state (None) operation. Precision for LG and LLL faults was found to be perfect (1.000). The minor variance observed (approx. 4.5%) occurs between the LL (Line-Line) and LLG (Line-Line-Ground) categories. This is a well-documented physical phenomenon where the initial sub-transient wave of an ungrounded inter-phase fault can momentarily mimic a ground return before the system neutralizes. This error does not impact protective relaying logic, as both states require an identical isolation response.

Classification Performance Metrics

The model achieved an aggregate accuracy of **98.19%**. Performance on critical Line-to-Ground (LG) and Three-Phase (LLL) faults was found to be perfect.

Table 2: Detailed Statistical Breakdown (N=2,040)

Fault Class	Precision	Recall	F1-Score	Support
LG (Line-Ground)	1.000	1.000	1.000	408
LL (Line-Line)	0.954	0.956	0.955	408
LLG (Double Line-Ground)	0.956	0.953	0.955	408
LLL (Three-Phase)	1.000	1.000	1.000	408
None (Steady State)	1.000	1.000	1.000	408
Model Aggregate	**0.982**	**0.982**	**0.982**	**2,040**

7 Operational Guide: Edge Deployment

Follow these steps to replicate the benchmarked **0.07 ms** performance in the field.

7.1 Hardware Initialization

Before launching the monitoring utility, lock the hardware clocks to prevent frequency scaling jitter:

```
sudo jetson_clocks
sudo nvpmodel -m 0
```

7.2 Monitoring Execution

Launch the fault classification binary using the optimized FP16 engines:

```
./fault_monitor   --engine="classifier_fp16.engine"
--dist-engine="locator_fp16.engine"
--input="/dev/grid_bus_0"
```

8 Conclusion

The technical validation of the **Fault Classification and Location model** confirms that advanced AI diagnostics can now operate at the same speeds as traditional silicon-based logic. By reducing inference latency to **0.07 ms** while maintaining a power draw of **992 mW**, we have created a production-ready solution for high-speed edge protection. This system provides the instantaneous, localized intelligence necessary to isolate faults before transients can escalate into grid-wide outages, ensuring the stability of critical energy infrastructure.