Harbin institute of Technology, Shenzhen
Md Ziad Bin Sorwar (2023331435)
Electronics and information engineering
Xiad79@gmail.com

Feature-Based Circuit Fault Diagnosis with Classical Machine Learning Baseline

Abstract

Circuit fault diagnosis is framed as supervised classification: from simulated AC sweeps and transient responses, the fault type and mode (R/C/L; open/short/drift) are predicted. The problem is interesting because models that perform well under nominal settings may generalize poorly when fault magnitudes, noise, or operating conditions shift. The proposed work builds a reproducible pipeline and compares feature-based learning with sequence models (1D-CNN and lightweight Transformer) to improve robustness.

1. Project Proposal

1) Problem. Robust multi-class fault classification will be studied under controlled distribution shift (train on mild faults; test on severe faults), connecting to course topics on CNNs/Transformers and optional RL.

2) Data and model structure. Data will be generated (not collected) using Simulate_Circuit.py (simulate_all, run_ac, run_transient) by sweeping RC/RL/RLC parameters and injecting faults via apply_resistor_fault/apply_capacitor_fault/apply_inductor_fault. Feature_Extraction.py (build_feature_dataset) will compute time/frequency descriptors (basic statistics, FFT peaks, bandwidth proxies). Data_Preprocessing.py (run_full_preprocessing_pipeline) will handle missing values, safe stratified train/val/test splits, standardization, and class-balancing weights, saving artifacts for reproducibility.

3) Method. Baselines in Train_Models.py (SVM, kNN, shallow NN, and a Keras classifier wrapper) will be reproduced first. Next, a 1D-CNN and a Transformer encoder will be added for raw-sequence learning. Experiments will be standardized using experiment_utils.py and run_experiment.py. As a stretch goal, an RL-style active-sensing policy will be explored to select informative frequency points or time windows.

4) Literature. Model-based diagnosis, time-series CNNs, Transformer sequence models, and RL for active sensing/experimental design will be reviewed.

5) Evaluation. Evaluation.py will report accuracy, macro-F1, balanced accuracy, and confusion matrices. Robustness will be assessed with performance-versus-shift curves over five or more seeds; paired t-tests or Wilcoxon signed-rank tests will compare models. Expected figures include class distributions, confusion matrices, and robustness curves.