Exploring Bloom Filters as Fault Detectors for Static Memory Content in Machine Learning Systems

Contacts

Elijah Cishugi - **daily supervisor** (e.s.cishugi@utwente.nl)
dr. ir. Marco Ottavi - **promoter** (m.ottavi@utwente.nl)

Background & motivation

Modern computing systems increasingly deploy **machine learning (ML)** models on edge devices, accelerators, and servers. These models often store large amounts of parameter data (**weights**) in memory. If this data becomes corrupted (e.g., due to bit flips, radiation, row/wordline faults, or intermittent memory errors), model accuracy can degrade, making **fault detection** important for reliability.

A **Bloom filter** is a compact probabilistic data structure used to test whether an element is a member of a set. It supports fast membership queries and is memory-efficient. Bloom filters have been explored in fault-detection contexts (e.g., as lightweight checkers for instruction memories [1, 2]). However, instruction memory content can change (updates, patches, dynamic loading, caching effects), which complicates how such detectors are constructed and maintained.

In contrast, many deployed ML scenarios have static memory content: after a model is loaded, its weights typically remain unchanged during inference. This makes ML weights a promising target for lightweight integrity monitoring techniques such as Bloom-filter-based detection.

Problem statement

Existing error correction/detection solutions (e.g., **ECC**, **CRC**) trade off coverage and overhead, vs implementation complexity. The key question is whether Bloom filters can serve as practical fault detectors for ML model weights when the protected content is static, potentially offering lower memory/compute overhead than traditional methods while still providing useful detection capability.

The **aim** is to design and evaluate Bloom-filter-based fault detection schemes for static ML weight memory, and compare them against baseline approaches in terms of:

- detection effectiveness under realistic fault models
- false-alarm behavior (false positive)
- memory and performance overhead
- scalability to modern model sizes

References

[1] E. S. Cishugi, T. T. Smit, B. Forlin, C. Cazzaniga, K.-H. Chen, and M. Ottavi, “Bloom filters for soft error detection: Neutron and fault injection validation,” in *2025 IEEE 31st International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2025, pp. 1–7.

[2] B. Forlin, E. B. Annink, E. Cishugi, C. Cazzaniga, P. Rech, G. Rauwerda, G. Furano, and M. Ottavi, “Neutron beam evaluation of probabilistic data structure-based online checkers,” in *2024 IEEE 30th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2024, pp. 1–6.