Ensemble learning based Architecture Vulnerability Factor calculation using partial feature set in processors

Jiabin Wang¹, Jiajia Jiao¹*, Yuzhuo Fu²
¹ Shanghai Maritime University, Shanghai, China
² Shanghai Jiao Tong University, Shanghai, China

Email: jiaojiajia@shmtu.edu.cn

Abstract. With the scaling technology, soft error induced bit upsets are increasingly threatening the processor reliability. Processor designers require effective tools or methodologies to estimate the often-used metric Architectural Vulnerability Factor (AVF). This paper presents an ensemble learning based AVF calculation methodology for fast reliability assessment. Instead of the entire feature set, only partial non-critical attributes are selected to build the predictive model so that many performance counters can be removed or shut down for saving memory space and power consumption. Millions of data collected from a cycle-accurate simulator sim-SODA, are trained by the latest learning methods in Tensorflow. The SPEC2000 results demonstrate the instanced ensemble learning-random forest and Ada-boost perform nearly perfect accuracy, better than linear regression, and neural network.

1. Introduction
The scaling technology makes soft error dominate in the computer reliability design. Due to the increasing error rate and complex patterns of multi-cell upsets, the more and more concern on soft error have received from the academic and industry. The designers of reliable processors should have effective reliability assessment methods or tools to support the dynamic configurations or early design guide.

Architectural Vulnerability Factor (AVF), as the often used metric, represents the probability of one bits upset occurring resulting in a wrong output in the defined observation point. It is usually evaluated by the fast Architecture Correct Execution (ACE) analysis [1] or accurate fault injection [2]. The former can make good use of the component utilization for the lower bound of AVF while the later adopts lots of time consuming simulations for characterizing the soft error impacts on processors. Few works also try some analytical models for fast evaluation of soft error influences [3]. To further meet the demands of online configurations for different benchmarks or architectures, the paper work aims at the machine learning methods applied into fast and accurate AVF predication.

2. Related work
Dynamic prediction of architectural vulnerability from micro-architectural state was proposed by identifying strong correlations between structural AVF values and a small set of processor metrics [4]. AVF stressmark was generated by an automated methodology for bounding the worst-case vulnerability to soft errors in [5]. The online estimation of architectural vulnerability factor for soft errors was implemented [6]. Unlike the previous work, the machine learning method of boosted
regression tree was used to predict micro-architectural vulnerabilities in [7]. A linear regression was used for predicting the detectable and unrecoverable errors (DUE\_AVF) and silent data corruptions (SDC\_AVF) [8]. Machine learning technique involved for off-line training of an application with representative inputs, and on-line detection using the model, applied even to a different dataset [9]. Machine learning based generic run-time method with low area and power overhead was presented to predict the soft-error vulnerability of on-chip memory arrays as well as logic cores[10]. The existing works with the chosen machine learning method perform well from the various experiments or verification. However, there is lack of comprehensive comparison to point the more appropriate learning approach in the AVF prediction.

As a complementary work, the paper compare the classical or hot learning methods (linear regression, regression tree, as well as neural network), we recommend an ensemble based cost-effective methodology to predict AVF using the partial performance metrics.

3. Proposed ensemble learning based AVF assessment methodology

3.1. Observation and analysis

A lot performance metrics can be collected as the attribute features. For example, the ACE analysis platform [11] provides about 190 metrics, e.g., sim\_num\_loads, simu\_num\_stores, sim\_num\_insns. To remove some invalid features, the principal component analysis (PCA) [12] is used in figure 1 to draw 10 features impacts on AVF. The distinct linear trend can be found for each SPEC2000 benchmark. However, the mixed results of all benchmarks show half linear region and half nonlinear region. Based on the observation, we can refer the linear regression can work in AVF prediction but not perfect due to the nonlinear factors.

3.2. Case study of ensemble learning for AVF prediction

Unlike interpretable linear regression, as a white box to represent the correlation of output and features clearly and directly, some linear based extensions are also attractive in the data analysis filed. Backward propagation neural network can be considered a special linear extension with nonlinear activation function (e.g., Sigmoid, ReLU, tanh). It works like a black box to handle the potential nonlinear factors. Ensemble learning like random forest and Xg-boost uses multiple week learning systems instead of a single strong learning system [13][14]. It seems a grey box, a trade-off between linear regression and neural network. It is referred that the ensemble learning works well in AVF prediction due to the grey box based compromise.

Random forest and Xg-boost are the represents of ensemble learning. The case study of random-forest for better understanding the ensemble learning is depicted in figure 2. The main idea of random
forest is to use regression tree for training each randomly selected data segment and sum up all of the weak training results as the final output.

Figure 2. Random forest: case study of ensemble learning.

4. Simulation and analysis

4.1. Simulation framework and configuration

To apply the machine learning methods into the AVF estimation, we build a complete simulation framework in figure 3. The overall framework combines the sim-SODA and tensorflow[15] together. Firstly, the millions of dataset are generated from SPEC2000 benchmarks execution in sim-SODA. The sampling interval is 1000 instructions for each benchmark. Secondly, the dataset are processed by data integration, feature selection and data clean respectively. Then, the clean data used for training and testing for different machine learning methods in tensorflow. Finally, two often-used metrics MAE and R2 are used for evaluation to decide the more appropriate approach.

Figure 3. Overall simulation framework of sim-SODA and tensorflow.
4.2. Results and analysis
The IQAVF (AVF for instruction queue) prediction can be evaluated by mean absolute error (MAE) and fitting excellence $R^2$. Local 30 features about instruction and data TLB are used for training. The evaluation results of MAE and $R^2$ for different learning methods comparison are listed in figure 4 and Table 1 respectively.

From the MAE results in figure 4, we can find out ensemble learning has nearly perfect mean absolute error. Instead, linear regression and neural network performs not well. Neural network is even worse than linear regression because of the linear trend in AVF estimation of subsection 3.1. This phenomenon is consistent with the theoretical analysis of learning ideas in subsection 3.2. Random forest and Ada-boost can takes full advantage of the multiple small week learning systems in the case of some local features training.

![Figure 4. MAE evaluation results comparison.](image)

From the perspective of fitting excellence $R^2$, the same conclusion can be drawn from Table 1. Linear regression and neural network have the abnormal results beyond the legal range $[0,1]$. It is reasonable for their powerlessness of using local features to predict the global AVF value. The recommended ensemble learning achieve the dominate win for a serial values of 0.999.

| Benchmark | Linear Regression | Neural Network | Random Forest | Ada-boost |
|-----------|-------------------|----------------|---------------|-----------|
| mcf       | 1.02E+08          | -68186.6       | 0.999896      | 0.999996  |
| gzip      | 0.835267          | -5.874        | 0.999999      | 0.999999  |
| twolf     | 15061664          | -11115.6      | 0.999346      | 0.999278  |
| crafty    | 13254025          | -200028       | 0.999898      | 0.999915  |
| econ      | 208803.8          | -2.2E+07      | 0.999979      | 0.999975  |
| gcc       | 955554.7          | -1.7E+10      | 0.990333      | 0.999673  |

In all, the above results demonstrate that the ensemble learning method is cost-effective and recommended for reliable processor design.
5. Conclusion
The paper presents an ensemble based cost-effective methodology to predict AVF using the partial performance metrics. The overall simulation framework of sim-SODA and tensorflow is built for comprehensive comparison. The results of both MAE and R2 show that the ensemble learning uses the selective 30 local features for 99.9% AVF prediction accuracy, better than linear regression and neural network.

References
[1]. Mukherjee S S, Weaver C, Emer J, et al. A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor[C]// Microarchitecture, 36th Annual IEEE/ACM International Symposium on. IEEE, 2003: 29-40.
[2]. Leveugle R, Calvez A, Maistri P, et al. Statistical fault injection: Quantified error and confidence[C]//Proceedings of the Conference on Design, Automation and Test in Europe. European Design and Automation Association, 2009: 502-506.
[3]. Nair A A, Eyerian S, Chen J, et al. Mechanistic modeling of architectural vulnerability factor[J]. ACM Transactions on Computer Systems (TOCS), 2015, 32(4): 11.
[4]. Walcott K R, Humphreys G, Gurumurthy S. Dynamic prediction of architectural vulnerability from microarchitectural state[J]. ACM SIGARCH Computer Architecture News, 2007, 35(2): 516-527.
[5]. Nair A A, John L K, Eeckhout L. AVF stressmark: Towards an automated methodology for bounding the worst-case vulnerability to soft errors[C]//Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2010: 125-136.
[6]. Li X, Adve S V, Bose P, et al. Online estimation of architectural vulnerability factor for soft errors[C]//ACM SIGARCH Computer Architecture News. IEEE Computer Society, 2008, 36(3): 341-352.
[7]. Li B, Duan L, Peng L. Efficient microarchitectural vulnerabilities prediction using boosted regression trees and patient rule inductions[J]. IEEE Transactions on Computers, 2010, 59(5): 593.
[8]. Wibowo B, Agrawal A, Tuck J. Characterizing the impact of soft errors across microarchitectural structures and implications for predictability[C]//Workload Characterization (IISWC), 2017 IEEE International Symposium on. IEEE, 2017: 250-260.
[9]. Liu J, Agrawal G. Soft Error Detection for Iterative Applications Using Offline Training[C]//High Performance Computing (HiPC), 2016 IEEE 23rd International Conference on. IEEE, 2016: 2-11.
[10]. Vijayan A, Kamehr S, Ebrahimi M, et al. Online Soft-Error Vulnerability Estimation for Memory Arrays and Logic Cores[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(2): 499-511.
[11]. Fu X, Li T, Fortes J. Sim-SODA: A unified framework for architectural level software reliability analysis[C]//Workshop on modeling, benchmarking and simulation. 2006.
[12]. Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and Intelligent Laboratory Systems, 1987, 2(1):37 – 52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geo-chemists.
[13]. Breiman L. Random forests[J]. MachineLearning,2001, 45(1):5–32.
[14]. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and DataMining, NewYork, NY, USA: ACM, 2016. 785– 794.
[15]. Abadi M, Barham P, Chen J, et al. Tensorflow: a system for large-scale machine learning[C]//OSDI. 2016, 16: 265-283.

Acknowledgments
This work is supported by National Natural Science Foundation, numbered 61502298, 61472244 and Innovation Program of Shanghai Maritime University.