

PERFORMANCE COMPARISON OF KNN AND LINEAR REGRESSION BASED MACHINE LEARNING APPROACHES IN THE DESIGN OF AN EM-AGEING AWARE SCHEDULER FOR IMPROVING THE LIFETIME OF MULTI-CORE PROCESSORS

P. Jagadeesh Kumar and M.G. Mini

Department of Electronics Engineering, Government Model Engineering College, India

Abstract

The increasing computing requirements in modern embedded systems demand high performance processing cores with highly down-scaled devices. System reliability due to ageing is a major concern in such processing cores. Run-time adaptations with ageing-aware schedulers are getting more attention to increase the lifetime reliability of the processing cores. Accurate and efficient thermal estimation of processing cores based on the characteristics of workloads is important to implement run-time adaptation schemes. In this work we have developed K-Nearest Neighbor (KNN) and Linear Regression (LR) machine learning models for the estimation of thermal profiles of the major power consuming logical units of a multi-core processor. Prediction performance of KNN and LR machine learning models in the design of an Electron Migration (EM)-ageing aware scheduler for improving the lifetime of multi-core processors is evaluated. The ageing-aware scheduler takes inputs from the trained models for the estimation of ageing effect and perform scheduling based on performance and reliability requirement. Experimental results show that predictive performance is better for KNN model compared to the LR model.

Keywords:

Ageing Aware Scheduler, Machine Learning Model, Multi-Core Processor, KNN, LR

1. INTRODUCTION

The emerging high-end embedded application areas such as automotive, 5G communication, networking etc. demand high performance computing systems. Dense integrated circuits with multiple cores are a solution to meet the high functionality per unit area requirements of such applications [1]. The highly down-scaled devices in densely packed integrated circuits operating at higher frequency and at elevated temperatures can accelerate the ageing effects such as electron migration, which brings down the useful lifetime of such computing systems [2]. Schedulers which can take appropriate task execution strategies at run time to minimize the ageing effects is a subject matter for many researchers [3]-[5]. Computationally efficient and accurate run-time power and thermal estimation techniques are essential for the implementation of such scheduling policies.

In this work, the proposed KNN and LR based machine learning models are used to estimate Steady State Temperature (SST) of the various logical units of the processor cores. A fine-grained approach, which is better suited to understand the localised power and temperature is followed. These estimated values are fed to the ageing aware scheduler for making run time scheduling decisions. The prediction accuracy of both the models are compared.

The paper is organised in the following order. In section 2, works related to power and thermal estimation are presented. In

section 3, the development of KNN and LR based models are deliberated. Tools used for the model development are also explained in section 3. Comparative analysis of KNN and LR model-based schedulers and results are discussed in section 4. Conclusion and future scope are covered in section 5.

2. LITERATURE REVIEW

The rising performance requirements in modern technology devices has inspired several researchers to focus on the development of efficient run time strategies for improving the lifetime reliability of processing cores. Haghbayan et al. [6] proposed a fine grained thermal-cycling aware Dynamic Reliability Management (DRM) approach for multi-core systems to mitigate ageing effects. A Negative Bias Temperature Instability (NBTI) aware task parallelism frame work for the execution of parallel tasks on multi-core systems in [7].

For the effective implementation of run time strategies, real time power and thermal profiles of the processor cores are a necessity. Thermal profiles of various logical units of the processing cores can be estimated with the knowledge of the power traces of corresponding logical units. HotSpot [8] is a widely used thermal modeling framework which takes the power trace file and floor plan file of the processing cores as inputs. Power traces of the logical units of processing cores can be estimated using Multicore Power, Area, and Timing (McPAT) which is an integrated power, area and timing modeling framework [9]. A detailed validation of McPATs power models is presented by Sam et al. [10]. McPAT needs execution statistics of workloads and micro-architectural parameters to estimate the power Figures of various units of the processor cores. The gem5 simulator, which is a configurable modular simulation framework can be used to evaluate the execution characteristics of workloads on processor architectures [11]. Gem5 supports most commercial ISAs (ARM, ALPHA, MIPS, Power, SPARC, x86 etc.) and different cache organisations and memory configurations can be defined for the analysis. Qureshi et al. [12] propose a gem-5 based system level simulation framework for the evaluation of architectures suitable for real-time video transcoding and image classification applications.

Use of the above-mentioned computer architecture simulators for real-time estimation of thermal and power profiles of the logical units is not suitable for run time adaptation of the computing system as it is in general computationally expensive. Computationally efficient and accurate models which can estimate the power and thermal profiles based on the workload characteristics is an important requirement for implementing run time strategies. Jaeckle et al. [13] and Takouna et al. [14] has explored the use of model-based techniques for the estimation of power and thermal profiles of processor cores. For computing

systems running typical embedded applications, machine learning models for the estimation of thermal and power profiles can be developed by closely analysing the execution characteristics of workloads. Machine learning-based temperature prediction for runtime thermal management across system components is proposed by Zhang et al. [15]. A real-time estimation approach for full chip transient heatmaps for commercial processors based on machine learning is proposed by Sadiq Batcha et al. [16]. A machine learning-based power and thermal management approach that dynamically learns the best encoder configuration and core frequency using information from frame compression, quality, performance, total power and temperature for High Efficiency Video Coding (HEVC) application is presented in [17]. Thus, a good number of related research works reported in recent years attempts to device multiple techniques for the estimation of power and thermal profile of processors, helpful in implementing various run-time strategies for improving processor life time.

Intrinsic failures related to processor wear-out happens during the useful life period in semiconductor products due to continuous usage at elevated temperature. Various wear-out mechanisms that reduce the processor useful lifetime include electron-migration and stress migration in interconnects, time dependent dielectric breakdown in gate-oxides, thermal cycling and cracking [18]. A runtime scheduler can effectively implement strategies to adapt processor characteristics for improving processor lifetime by predicting power and thermal profiles with the help of machine learning models.

3. PROPOSED WORK

In this work we have developed KNN and LR based machine learning models for the estimation of steady state temperature of the major power consuming units of processing cores. The proposed methodology is suitable for processing systems running embedded applications as there is a high degree of predictability in various characteristics such as the number and type of instructions executed to complete specific operations, the type of data processed, characteristics of memory related operations etc. of workloads. We use KNN and LR regression analysis to infer and form the relationship between the workload characteristics and power and thermal profiles of the various logical units of processing cores.

Various tasks in MiBench suite, which comes under different embedded applications areas such as consumer, network and security, are considered as representative applications in this work. MiBench is an open-source benchmark, with a set of 35 embedded applications organised into six suites, and each suite targets a specific area of embedded application for benchmarking purpose [19]. Gem5 simulator is used to analyse the characteristics of representative MiBench bench mark tasks executing on a hexa-deca homogeneous multicore architecture. McPAT tool is used to compute power consumption of various logical units in the processor architecture corresponding to estimated workload characteristics. Power traces of the logical units of processor cores corresponding to workload characteristics along with chip and packaging specifications are fed to the tool HotSpot for computing thermal profile. We propose two regression methodologies, KNN and LR, to infer the association between workload characteristics and the corresponding power

and thermal profiles to develop machine learning models. These developed models will be helpful in run time estimation of power and thermal profiles once fed with the workload characteristics.

The representative applications from the MiBench suite, JPEG encode/decode, Dijkstra and Secure Hash Algorithm (SHA), whose execution characteristics on multiple processor configurations are used to develop data set for training the models. JPEG task implements encoding and decoding algorithms for image compression and decompression that focuses on multimedia applications and comes under consumer devices category in MiBench suite. The Dijkstra benchmark comes under network category, uses repeated application of Dijkstra's algorithm to calculate the shortest path between every pair of nodes in a graph and represents embedded applications in network devices such as in routers and switches. The Secure Hash Algorithm (SHA) comes under data security in MiBench suite and is an important consideration in e-commerce related applications. SHA benchmark in MiBench suite is a secure hash algorithm which produces a 160-bit message digest for the given input.

We use Syscall Emulation mode (SE mode) of gem5 to trap and emulate system calls executed by the workloads for analysing the following characteristics: number of CPU cycles simulated, number of idle cycles, number of busy cycles, number of integer/float instructions, number of load/store instructions, number of integer register/floating register read/write operations and number of cache read/write hits/misses. A hexa-deca homogeneous multicore having x86 architecture-based processing cores with two level cache hierarchy is selected as the computing platform.

3.1 KNN REGRESSION MODEL

KNN is a supervised learner for both classification and regression. KNN regression is a prediction task based on similarity measure, in which the target variable is numeric. We use the regressor 'sklearn.neighbors.KNeighborsRegressor' available from Scikit-learn, where it takes the mean of the k -nearest neighbors to predict the target element [20]. Workload characteristics that are having higher influence in the power consumption of a logical unit are identified and used as the training data set for developing the thermal model of a logical unit. Steady state temperature model of a logical unit is derived using dynamic power consumption of the logical unit and micro architectural parameters of the target processor.

3.2 LINEAR REGRESSION MODEL

In linear regression approach, steady state temperature of a logical unit is modeled as a linear combination of dynamic power and the micro-architectural parameters of the target processor. The predicted value \hat{y} of the model which is trained with a set of features $X = (x_1, x_2, \dots, x_p)$, is represented as in Eq.(1)

$$\hat{y}(w, x) = w_0 + w_1x_1 + \dots + w_px_p + b \quad (1)$$

where w_1, w_2, \dots, w_p represent the coefficients and b is the intercept term called the bias. Linear regression fits a linear model with coefficients $W = (w_1, w_2, \dots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. In order to quantify the goodness of the fit, a loss function $L(\hat{y}, t)$ is defined which denotes

how far off the prediction \hat{y} is from the target t . The loss function $L(\hat{y}, t)$ is represented as in Eq.(2)

$$L(\hat{y}, t) = 0.5(\hat{y} - t)^2 \quad (2)$$

where the value $(\hat{y} - t)$ is known as the residual. The goal is to choose w_1, w_2, \dots, w_p and b to minimize the loss function where

$$\begin{aligned} E(w_1, w_2, \dots, w_p, b) &= \frac{1}{N} \sum_{i=1}^N L(y^i, t^i) \\ &= \frac{1}{N} \sum_{i=1}^N \left(\sum_j w_j x_j^i + b - t^i \right)^2 \end{aligned} \quad (3)$$

Linear regression will take in its fit method arrays x, y and will store the coefficients of the linear model in its coefficient member W .

3.3 EM-AWARE SCHEDULER

The ever-increasing performance requirements of modern computing devices augmented with advancements in technology scaling allows chip designers to squeeze greater number of transistors per unit area. Such densely integrated processors when operated at higher clock rates to meet the performance demands, instigate various silicon wear-out issues during its useful lifetime. Electron-migration is one of the dominant wear-out mechanism in such integrated circuits. The key challenge is to predict the device degradation based on the application characteristics.

In this work we assume the useful lifetime of the processing cores to be 10 years, when operated with a junction temperature T_J of 105°C , by referring to the data available for current generation of Texas Instruments industrial grade embedded processors [21]. Assuming the device is operated within the specified data sheet voltage and frequency, the critical variable influencing silicon lifetime will be the junction temperature (T_J) of silicon. The processor lifetime is normalized using an Acceleration Factor (AF) and we apply Arrhenius equation to compare the wear-out due to electron migration accumulated over time for different operating temperatures. The acceleration factor is represented in Eq.(5)

$$AF = \exp\left(\frac{E_a}{K} \left(\frac{1}{T_{use}} - \frac{1}{T_{stress}}\right)\right) \quad (5)$$

where,

AF = Acceleration Factor,

E_a = Activation energy in eV,

K = Boltzmann's constant ($8.63 \times 10^{-5} \text{eV/K}$),

T_{use} = Use temperature in K ($C + 273$),

T_{stress} = Stress temperature in K ($C + 273$),

The proposed scheduler which takes run time decisions to minimize processor ageing due to electron migration, estimates the steady state temperature using the mentioned KNN and LR machine learning models. In this work we evaluate the performance of the machine learning models in the estimation of steady state temperature of various logical units of the processor cores. A fine-grained approach is followed where the lifetime of the processor is estimated at the granularity of the functional units on chip. With the estimated thermal profile, the scheduler's objective is to perform run-time mapping of applications to the processor cores such that the lifetime of the chip is maximized

while satisfying the performance requirement constraints. Block level representation of the proposed logic is represented in Fig.1.

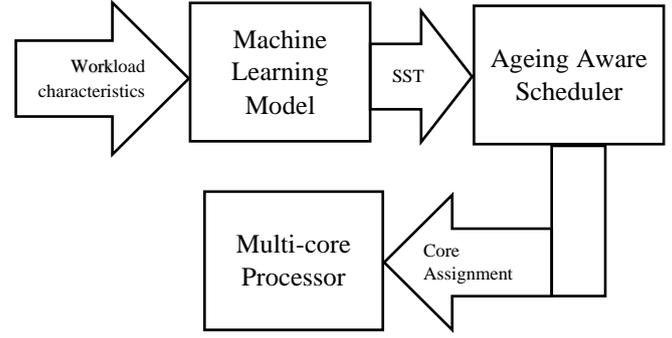


Fig.1. Processor lifetime enhancement scheme

For the estimation of operating frequency of processor cores, we attempted linear interpolation scheme, to construct new frequency points within the range of known data frequency points. The interpolation scheme is needed for the finer estimation of core operating frequency, as the machine learning models can estimate the temperature of the logical units of the processor core that is operating with a particular frequency from a discrete set of possible operating frequencies. The data set consists of estimated steady state temperature values of the logical units, corresponding to different frequencies of operation of the processor core. We use the `interp1d` class in `scipy.interpolate` module to create a function based on fixed frequency data points for implementing interpolation [22]. In the case of linear interpolation, linear polynomials are used to construct new data points within the range of a discrete set of known data points. The pseudo-code of the scheduler is represented in Algorithm 1.

Algorithm 1: Ageing Aware Scheduler Logic

Input: Workload Characteristics

Output: Mapping of the workloads $\{W_1, W_1, \dots, W_n\}$ to cores $\{C_1, C_1, \dots, C_m\}$

Step 1: While (true)

Step 2: For each scheduling interval T_s do {

Step 3: For each workload W_i in $\{W_1, W_1, \dots, W_n\}$ in the service queue S do

{

Step 4: Estimate thermal profiles of the feasible set of processors

Step 5: Cores C_i using machine learning models.

Step 6: Estimate ageing factor of the cores selected

Step 7: Decide the frequency of operation of the selected core

Step 8: }//end for each workload

Step 9: }//end for each scheduling interval

Step 10: }//end while.

4. RESULTS AND DISCUSSION

The experiments were conducted using four applications from MiBench suite, JPEG encode/decode, Dijkstra and Secure Hash Algorithm (SHA) belonging to consumer, network and data security applications respectively. Benchmark applications are compiled and statically linked with glibc using the open-source

compiler gcc [23]. The execution characteristics of the workloads are profiled using gem5. The corresponding dynamic power and steady state temperature are estimated using McPAT and HotSpot respectively. We have used multiple instances of the applications for developing the machine learning models for steady state temperature estimation. The data set processed by the applications are varied, thus deriving n instances of a workload w, and profiled the execution characteristics of these workloads on m cores operating at different clock frequencies. Machine learning models are developed for the major power consuming logical units of the processor cores, using the workload characteristics and the corresponding power and thermal profiles. The major power consuming logical units in the execution unit of the processor cores are identified as integer register file, floating point register file, integer ALU, floating-point unit, load/store queue, data translation lookaside buffer (DTLB) and instruction translation lookaside buffer (ITLB). Developed models are validated by estimating the power and thermal profiles of the logical units using new instances of the workloads.

operating with a clock frequency of 3.4GHz. The corresponding percentage error in the estimation of the steady state temperature of logical units is presented in Fig.3.

Validation of the thermal model in the estimation of steady state temperature of the logical unit: int_ALU, for different tasks under consideration is represented in Fig.4. The corresponding percentage error in the estimation of the steady state temperature of the logical unit is presented in Fig.5.

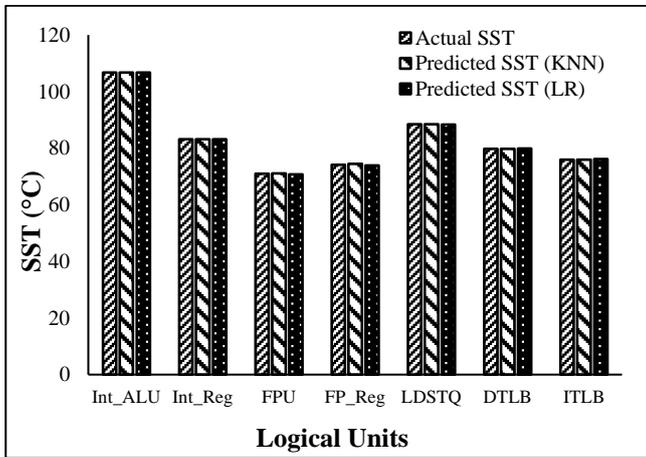


Fig.2. Comparison of the SST values of the logical units for the task djpeg

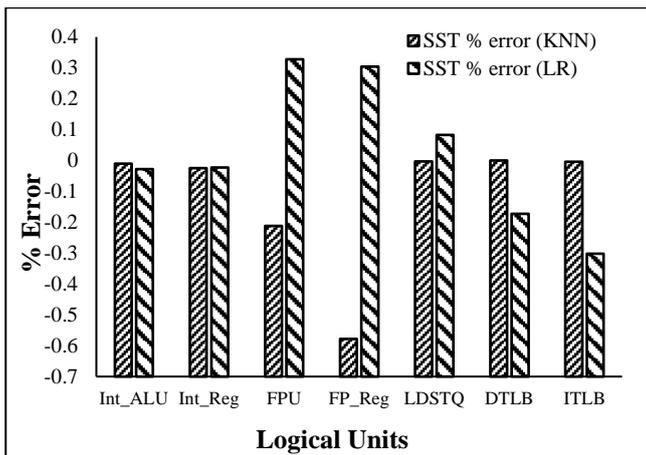


Fig.3. Percentage error in the estimation of SST values of the logical units for the task djpeg

The Fig.2 shows the validation of thermal models by comparing the estimated values of the steady state temperature using KNN and LR models with that estimated using HotSpot, for the application djpeg, processing a new dataset, when the core is

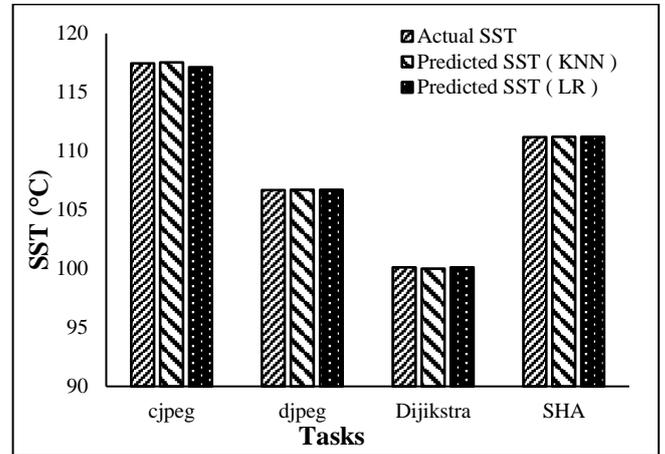


Fig.4. Comparison of the SST values of int_ALU for different tasks

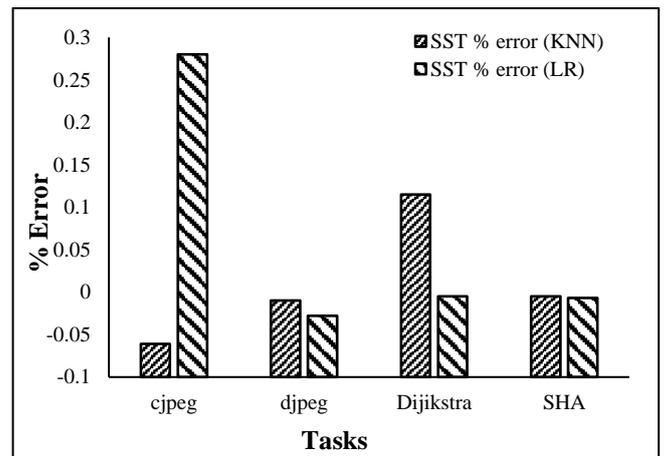


Fig.5. Percentage error in the estimation of SST of int_ALU for different tasks

In the proposed fine-grained approach, for a task that is ready for execution, steady state temperature of the major power consuming logical units of the processor core are estimated using machine learning models. The Table.1 summarizes the Root Mean Square Error (RMSE) values in the estimation of steady-state temperature of the major power consuming logical units.

Table.1. RMSE values of the estimated SST

Task		cjpeg	jpeg	Dijkstra	SHA
int_ALU	RMSE(KNN)	0.058	0.009	0.103	0.006
	RMSE(LR)	0.45	0.023	0.003	0.005
int_Reg	RMSE(KNN)	0.095	0.015	0.057	0.004

	RMSE(LR)	0.711	0.014	0.003	0.004
FPU	RMSE(KNN)	0.02	0.52	0.03	0.003
	RMSE(LR)	0.284	0.229	0.098	0.003
FP_Reg	RMSE(KNN)	0.134	0.248	0.036	0.004
	RMSE(LR)	0.374	0.427	0.111	0.003
LDSTQ	RMSE(KNN)	0.45	0.017	0.063	0.003
	RMSE(LR)	0.722	0.065	0.003	0.004

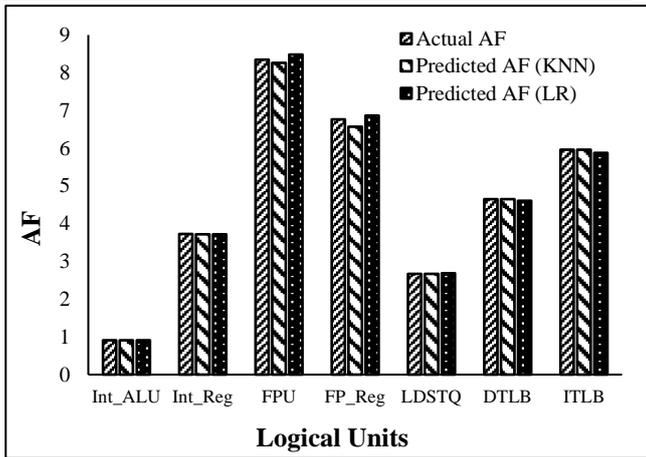


Fig.6. Comparison of AF values of the different logical units for the task djpeg

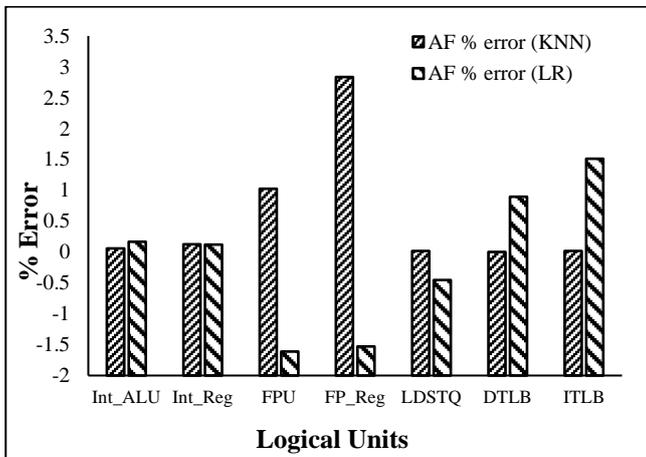


Fig.7. Percentage error in the estimation of AF of the different logical units for the task djpeg

The proposed ageing aware scheduler computes acceleration factor due to ageing of the cores for the ready tasks. Scheduler uses the SST values that are estimated with machine learning models for the computation of AF. Fig.6 compares AF of major power consuming units estimated using the machine learning models with the actual values. Percentage error in the estimation of AF of the different logical units when the task djpeg is scheduled on a core running at 3.4GHz clock is represented in Fig.7.

The estimated AF values of the highest power consuming logical unit: int_ALU is compared with the actual values of AF in

Fig.8. Percentage error in the estimation of AF values are represented in Fig.9.

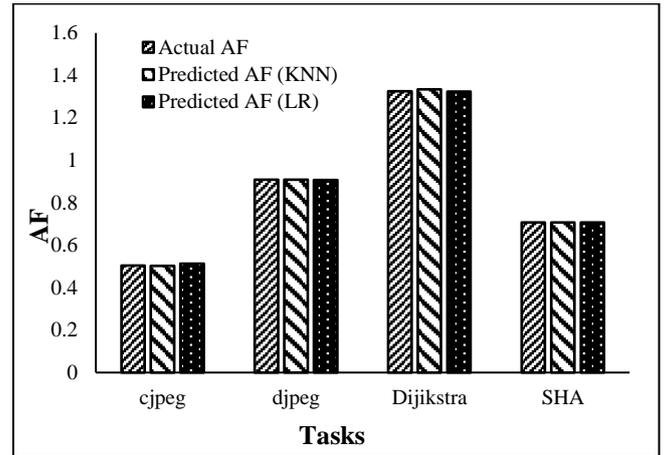


Fig.8. Comparison of AF values of int_ALU for different tasks

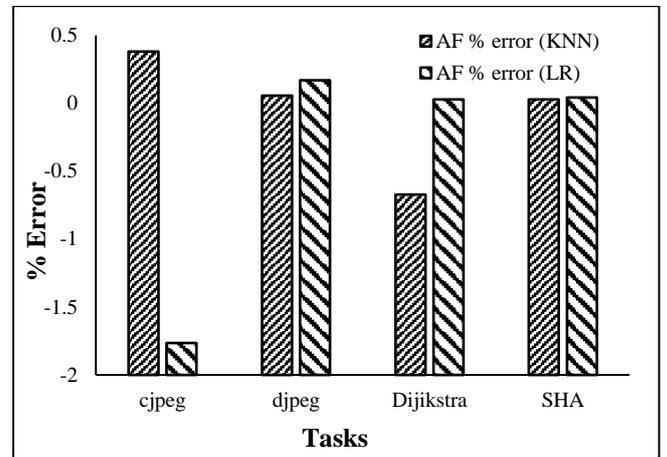


Fig.9. Percentage error in the estimation of AF of int_ALU for different tasks

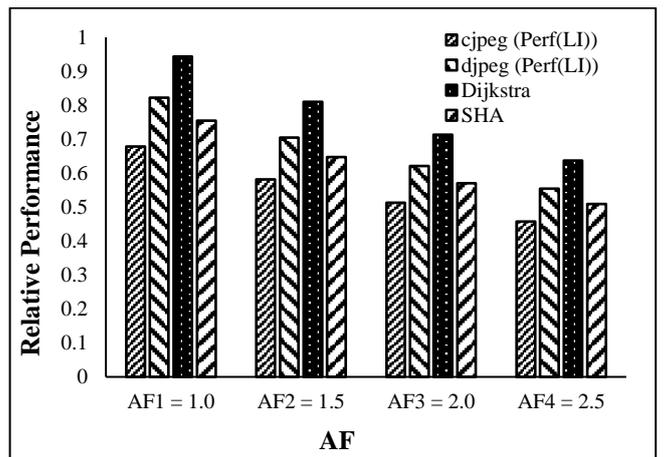


Fig.10. AF-Performance tradeoff (Actual AF with Linear Interpolation)

The AF-performance trade-off of the applications are evaluated when KNN and LR machine learning models are used for the estimation of steady state temperature. The frequency data point corresponding to the estimated temperature is constructed

using linear polynomial. The Fig.10 represents the AF-performance trade-off of the bench mark tasks, when AF computations are based on the actual values of the steady state temperature. The Fig.11 and Fig.12 shows the AF-performance trade-off of the bench mark tasks, when AF computations are based on the steady state temperature predicted using the proposed machine learning models.

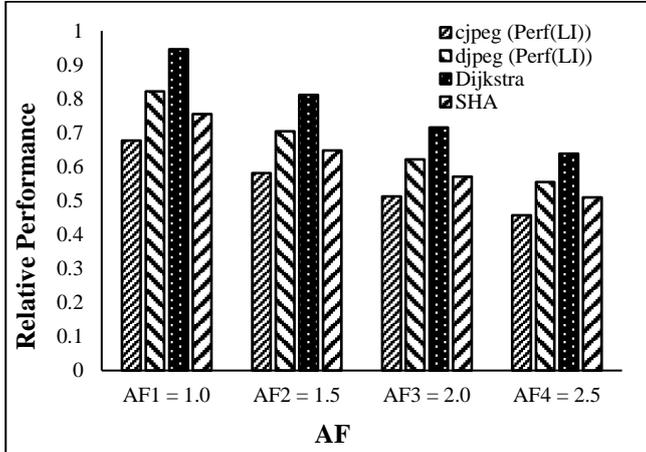


Fig.11. AF-Performance tradeoff (KNN with Linear Interpolation)

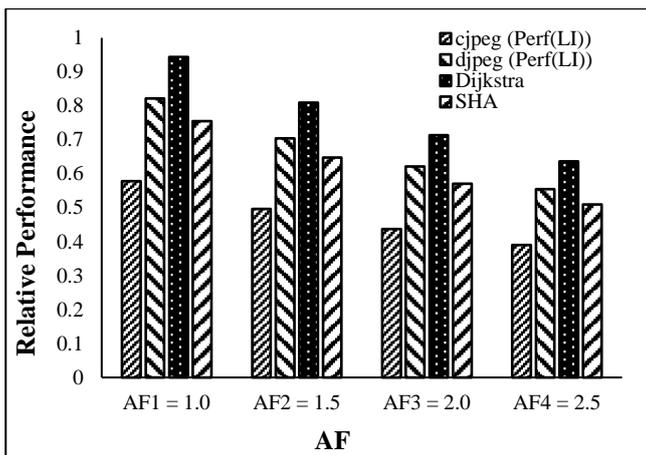


Fig.12. AF-Performance tradeoff (LR with Linear Interpolation)

5. CONCLUSION AND FUTURE WORK

In this work we have evaluated the performance of KNN and LR machine learning models in the estimation of thermal profiles of the processing cores, based on which an ageing-aware scheduler can take decisions to adapt the processor characteristics and improve the life-time reliability. The proposed ageing aware scheduler estimates the frequency of operation of the cores for meeting the life-time reliability budgets, with a graceful degradation in performance. The AF-performance characteristics of the scheduler based on the proposed machine learning models are closer to the one which uses the actual temperature values.

The performance of the machine learning models can be further improved, as it is possible to update the models periodically using the data from on-chip sensors. The scope of the scheduler can be extended by including the effects of ageing due to time dependent dielectric breakdown in gate-oxides, negative-

bias temperature instability, thermal cycling and cracking, etc. along with electron migration.

REFERENCES

- [1] V. Rajaraman, "Multi-Core Microprocessors", *Resonance*, Vol. 22, pp. 1175-1192, 2017.
- [2] J. Srinivasan, S.V. Adve, P. Bose and J.A. Rivers, "The Case for Lifetime Reliability-Aware Microprocessors", *Proceedings of Annual International Symposium on Computer Architecture*, pp. 276-287, 2004.
- [3] Y.G. Kim, M. Kim, J. Kong and S.W. Chung, "An Adaptive Thermal Management Framework for Heterogeneous Multi-Core Processors", *IEEE Transactions on Computers*, Vol. 69, No. 6, pp. 894-906, 2020.
- [4] P. Mercati, F. Paterna, A. Bartolini, L. Benini and T.S. Rosing, "WARM: Workload-Aware Reliability Management in Linux/Android", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 36, No. 9, pp. 1557-1570, 2017.
- [5] A. Sabu, B. Raveendran and R. Ghosh, "SMILEY: A Mixed-Criticality Real-Time Task Scheduler for Multicore Systems", *Proceedings of International Symposium on Distributed Simulation and Real Time Applications*, pp. 1-5, 2018.
- [6] M.H. Haghbayan, A. Miele, Z. Zou, H. Tenhunen and J. Plosila, "Thermal-Cycling-aware Dynamic Reliability Management in Many-Core System-on-Chip", *Proceedings of European Workshop and Exhibition on Design, Automation*, pp. 1229-1234, 2020.
- [7] Y. Chen, Y. Lin and I. Lin, "An NBTI-aware Task Parallelism Scheme for Improving Lifespan of Multi-core Systems", *Proceedings of International Symposium on Quality Electronic Design*, pp. 117-122, 2020.
- [8] R. Zhang, M.R. Stan and K. Skadeon, "HotSpot6.0: Validation Acceleration and Extension", Technical Report, Available at https://www.cs.virginia.edu/~skadron/Papers/HotSpot60_T_R.pdf, Accessed at 2015.
- [9] Li Sheng, Ho Jung and R.D. Strong, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures", *Proceedings of International Symposium on Microarchitecture*, pp. 12-16, 2009.
- [10] L. Sam, J. Hans and B. Pradip, "Quantifying Sources of Error in McPAT and Potential Impacts on Architectural Studies", *Proceedings of International Symposium on High Performance Computer Architecture*, pp. 577-589, 2015.
- [11] Bradford Beckmann and Gabriel Black, "The Gem5 Simulator", *ACM SIGARCH Computer Architecture News*, pp. 1-7, 2011.
- [12] Y.M. Qureshi, W.A. Simon, M. Zapater, D. Atienza and K. Olcoz, "Gem5-X: A Gem5-Based System Level Simulation Framework to Optimize Many-Core Platforms", *Proceedings of International Conference on Simulation*, pp. 1-12, 2019.
- [13] D. Jaeckle and A. Sikora, "Thermal Modeling of Homogeneous Embedded Multi-Core Processors", *Proceedings of International Conference on Advances in*

- Computing, Communications and Informatics*, pp. 588-593, 2014.
- [14] I. Takouna, W. Dawoud and C. Meinel, "Accurate Multicore Processor Power Models for Power-Aware Resource Management", *Proceedings of International Conference on Dependable, Autonomic and Secure Computing*, pp. 419-426, 2011.
- [15] K. Zhang, "Machine Learning-Based Temperature Prediction for Runtime Thermal Management Across System Components", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 29, No. 2, pp. 405-419, 2018.
- [16] S. Sadiqbatcha, Y. Zhao, J. Zhang, H. Amrouch, J. Henkel and S.X. Tan, "Machine Learning Based Online Full-Chip Heatmap Estimation", *Proceedings of Asia and South Pacific Conference on Design Automation*, pp. 229-234, 2020.
- [17] A. Iranfar, M. Zapater and D. Aienza, "Work-In-Progress: A Machine Learning-Based Approach for Power and Thermal Management of Next-Generation Video Coding on MPSoCs", *Proceedings of International Conference on Hardware/Software Codesign and System Synthesis*, pp. 1-2, 2017.
- [18] J. Srinivasan, S. V. Adve, P. Bose and J.A. Rivers, "The Case for Lifetime Reliability-Aware Microprocessors", *Proceedings of Annual International Symposium on Computer Architecture*, pp. 276-287, 2004.
- [19] M.R. Guthaus, J.S. Ringenberg, D. Ernst, T.M. Austin, T. Mudge and R.B. Brown, "MiBench: A Free, Commercially Representative Embedded Benchmark Suite", *Proceedings of 4th Annual IEEE International Workshop on Workload Characterization*, pp. 3-14, 2001.
- [20] F. Pedregosa, "Scikit-Learn: Machine Learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, 2011.
- [21] Allan Webber, "Calculating Useful Lifetimes of Embedded Processors", Texas Instruments Application Report, pp. 1-16, 2020.
- [22] P. Virtanen, R. Gommers and T.E. Oliphant, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nature Methods*, Vol. 17, pp. 261-272, 2020.
- [23] Richard M. Stallman and the GCC Developer Community, "Using the GNU Compiler Collection", Available at: <https://gcc.gnu.org/onlinedocs/gcc.pdf>, Accessed at 2021.