IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Failure prediction in Embedded Systems using Long Short-Term Inference

Dr. S. Mary Praveena
Department of Electronics and Communication Engineering
Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

M.Abijith
Department of Electronics and Communication Engineering
Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

M.Akash
Department of Electronics and Communication Engineering
Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

S.Karthick
Department of Electronics and Communication Engineering
Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

H.Kumaravel
Department of Electronics and Communication Engineering
Sri Ramakrishna Institute of Technology, Coimbatore, Tamil Nadu, India

Abstract—Users of embedded systems and cyber-physical systems expect dependable operation for diverse sets of applications in environments. Reactive self-diagnosis techniques either use unnecessarily old preventive methods, or it does not prevent the system from failure. In this paper, we use the MACHINE LEARNING(ML) techniques to prevent the systems from those catastrophizes. we evaluate and analyse the alternative using machine learning techniques for predicting temperature behaviour on a mobile system-on-chip, and propose realizable hardware.

INTRODUCTION

The dependency on embedded systems is increasing wisely. These systems are combined to build much-complicated hardware to extract the predicted outcome. So, the demand for the embedded system software and the application they depend on are also increasing wisely. Such a system should be designed in such a manner that they are dealing with a large number of internal and external variabilities, threats, and uncertainties in their lifetimes. So, it should predict the error before it actually occurs.

So, to maximize the lifetime of the particular system(hardware), the self-diagnosed process was initiated at an early stage in order to identify the degradation and immediate errors that occur. These techniques can be combined with unsupervised platform self-adaptation to meet performance and safety targets. Self-diagnosis techniques that are reactive may

(a) not be sufficient to address catastrophic failures, or

(b) take overly conservative approaches that block the performance.

For instance, consider the technique called thermal management of embedded system-on-chip. This system (SoC) controls heat in the system one is to define the temperature of the system and another one is to throttle. If the throttle exceeds this SoC prevents the system from being overheated. As stated, if the temperature has been monitored, they could come across with the protective behaviour in advance to protect the system. However, without workload knowledge temperature is unpredictable as it is nonlinear.

Neural networks, one of the types of machine learning techniques are useful for identifying complex system dynamics. However, neural networks are complex and difficult to deploy on power-constrained embedded systems. In this paper, we propose a failure prediction technique for embedded systems using long short-term memory (LSTM), LSTM networks are well suited to classifying, processing, and making predictions based on time series data, we demonstrate the effectiveness of our predictor for predicting temperature behaviour with respect to a threshold on an ODROID-XU3 platform, which is a single-board computer for multi-processing. This is a candidate for mitigating overheating failures and implementing efficient control policies. We specify an implementation that is realizable in hardware on low-power embedded systems. The specific contributions are as follows:

We propose a method for hardware failure prediction called Long Short-Term Prediction Model. This paper propose an architecture and hardware implementation of non-intrusive prediction engine based on the Long Short-Term Prediction Model to predict temperature behaviour in embedded systems. This paper evaluate the predictor using measured temperature data from an ODROID XU-3.

PROPOSED ALGORITHM

2.1. Background and Related Work

When modern systems-on-chip (SoCs) operate for extended periods, power dissipates. This can increase the temperature that impacts the chip. If we can control these thermal management processes, we can protect against failures in advance. A number of strategies have been proposed for onchip thermal prediction, and the methods can be classified into two steps.

The first prediction method builds models based on the measured temperature and power consumption. The second method builds the prediction model indirectly using equations, without thermal measurements. When adequate sensors are implemented in the systems, with the help of machine learning we can extract the data required from any complex architecture. Failure prediction has been proposed using support vector machines (SVMs), which are used to classify the data for analysis. Convolutional neural networks (CNNs) and a combination of techniques.

RNNs are naturally suited for learning temporal sequences and modelling time-series behaviours. RNNs have been applied to predict various behaviour in a large scale. In [6], the authors compare an RNN solution with an LSTM solution and observe that LSTMs significantly outperform RNNs in terms of accuracy.

In [2],[11] LST Measured in other domains for times series predictions such as water quality estimation, stock transaction prediction, mechanical states, and more. The authors compare LSTM networks with alternatives such as backpropagation neural networks, online sequential extreme learning machines, and support vector regression machines (SVRM), and demonstrate the superiority of LSTMs.

2.2. Contributions

The paper propose a method for predicting runtime behaviour in hardware: the Long Short-Term Prediction Engine. predicting runtime temperature behaviour on an embedded system-onchip. Our goal is to predict temperature behaviour so that, the temperature behaviour could be calculated well and advanced and it could be avoided, with a solution that can feasibly be integrated into an embedded System-on-Chip. Our SoC consists of four ARM (processors) A15 cores, with a shared L2 cache connected via bus. We measure the total power and temperature of the entire core cluster as per how much it has been utilized. To generate workloads, we use a synthetic microbenchmark [12] that is configurable. The microbenchmark (is a program is to measures the specific or single task) is able to stress the architecture in a wide range and we generated a "general-purpose" workload by executing the microbenchmark in phases that exercised different behaviour in these various dimensions. We execute different sequences on multiple cores to emulate different applications to train the model and test its performance. The prediction engine consists of two parts: a short-term binary model and a long-term regression model. The short-term binary model makes precise predictions quickly, useful for subtle changes, i.e., anticipating violations of a temperature threshold. The long-term regression model can make a prediction further in advance, useful to predict general behaviour in less-critical scenarios, i.e., predicting temperature trends in a safe state.

$$average(t)=\frac{1}{n}\sum_{i=0}D(t-i)$$

$$max(t)=MAX\{D(t-i),D(t-i+1),...,D(t)\}$$

$$min(t)=MIN\{D(t-i),D(t-i+1),...,D(t)\}$$

$$amplified(t)=\frac{D(t)-average(t)}{max(t)-min(t)}\times 100$$

2.2.1. SHORT TERM BINARY MODEL

The short-term binary model is used to predict unwanted behavior i.e., constraint violation. In our case in which we have a temperature threshold, we do not want to violate it; the shortterm binary model is utilized when the measured temperature is nearing the threshold. In this scenario, a slight rise in temperature will cause a failure (violation of constraint), thereby it is important to have a high recall rate. The recall rate must be tuned carefully to balance accuracy and overhead.

1) Model Definition

The short-term binary model is defined as follows:

- Input: temperature, core utilization, power
- Output: probability of failure (after a specific threshold, the model produces a binary result: '0' refers to normal and number '1' refers to failure)

2) Model Training

This paper first isolate the data above the critical point (85°C) to use as training data. Because the range of the data is reduced, we amplify the changes of data to increase its variation. When performing amplification at runtime, we must consider constraints such as the real-time hardware implementation and the short failure intervals. We create a method called Sliding Average Amplification to efficiently preprocess data in order to increase variation and applied it on the four features. The method takes local data (5 timesteps) and uses Min-Max Normalization to amplify the values. The following equations show the calculation of Sliding Average Amplification. D(t) refers to the feature value at *t* and *i* refers to the number of timesteps defined as local data. .

3) Improved Loss Function

$$Loss=-(\alpha y\log\hat{}y+(1-\alpha)(1-y)\log(1-y\hat{})) \qquad (5)$$

$$\alpha=0.992 \qquad (6)$$

4) Model Structure

The simplest structure of an RNN prediction model that provides the required accuracy in order to minimize the hardware overhead.

The LSTM (Long Short-Ter Memory) internal structure is defined in the following equations. *x* indicates the input features, *h* is the output result, *W*, *b* are weights and bias and *c* is the intermediate variables. (7) (8) (9)

$$i_t=\sigma(W_{xi}x_t+W_{hi}h_{t-1}+b_i)$$
$$f_t=\sigma(W_{xf}x_t+W_{hf}h_{t-1}+b_f)$$
$$o_t=\sigma(W_{x0}x_t+W_{h0}h_{t-1}+b_0)$$
$$\tilde{c}_t=tanh(W_{xc}x_t+W_{hc}h_{t-1}+b_c)$$
$$c_t=f_t\odot c_{t-1}+i_t\odot\tilde{c}_t$$
$$h_t=o\odot tanh(c_t)$$

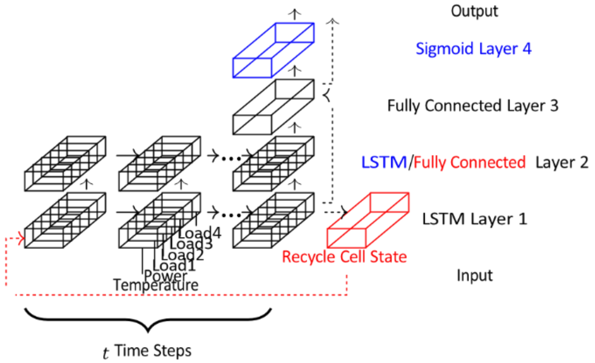

Fig. 3: Integrated model structure. The structures are shared between the short-term binary model and the long-term regression structure, depending on which is active.

Functionality and structure specific to the short-term binary model are in blue, and specific to the long-term regression model is red.

2.2.2. LONG TERM REGRESSION MODEL

The long-term regression model is used to predict behaviour in the normal state. The temperature varies in a large range depending on how the system is being used, in this state. Our main goal is to predict the temperature in advance to make run-

decisions in order to avoid critical states and also to optimize performance. It is necessary to ensure whether the prediction engine can be applied during normal execution to avoid critical states without sacrificing performance. As the system state is non-critical, precision can be sacrificed for universality. To this end, we build a regression model for long-term prediction.

1) Model definition

- Inputs: temperature, power, per-core utilization
- Outputs: temperature

2) Model Training

This paper use a larger range of training data (60-85◦C). The temperature variation occurs generally due to changes in the operating frequency and in the utilization of the core. We categorize training workloads as follows: unicore, multicore, and shifting. We execute combinations of synthetic benchmarks to compose our workloads. The benchmarks show variations in instructions-per-cycle (IPC), utilization, and cache miss rate, exercising the processor in a wide range.To achieve unicore workloads, we've to run every benchmark on one core to maintain a stable workload state. And then the multiple benchmarks are combined and start one after another to change the workload state on particular core. For multicore, different benchmarks were assigned on different cores and simultaneously start them. The workloads are maintained by assigning the different benchmarks on different cores and started at different times.

The data collected from the ODROID-XU3 does not appear stable initially, making essential filtering. The filters are used to smooth the raw data and consider the feasibility of hardware, then it is concluded that the data were preprocessed by recursion average filter that provides the most accurate model. Sizes of filters of each input were accurately determined.

3) Model Structure

LSTM is generally used to store Long-term memory, therefore, to work with long-term cases, LSTM is chosen for the structure for our model. When compared to a short-term model, longrange historical data is used to predict the accurate large temperature range. This leads to an increase in model time and execution time. Hence LSTM theory is used in the cell structure, letting the output cell state as the initial state.

2.2.3.HARDWARE IMPLEMENT FRAMEWORK

By integrating the short- and long-term models, we designed a single shared-hardware that supports all of Figure 3. With help of a sensor, the judgment module receives a temperature value and decides which model to activate. The short-term prediction model has activated If the temperature is greater than 85◦C. If it is less than 85◦C then the long-term prediction model is activated.

The structural overhead is reduced by connecting, the core LSTM and fully connected layers are shared partially, composed with the least common parameters (LSTM: 8-time steps, 64 hidden layers; fully connected: 4 hidden layers). The excess time steps can be stored in a state buffer and feedback

Using the implementation of LSTM we can calculate the

12960 FFs, 7201 LUTs, and 16 BRAM overhead. The LSTM hardware is 20 times faster than the Zync ZC7020 ARM-based hard-core processor (4.4 µs per inference), 44 times more power-efficient.

2.3 EVALUATION

The effectiveness of both the Short-Term Binary Model and Long-Term Regression Model is evaluated separately, with the help of additional data measured from the ODROID-XU3. These measured data consist of model input data measured at 5ms intervals. We perform sensitivity analyses of LSTM/RNN models for calculating different parameters and structures.

A. Short-Term Binary Model Evaluation

1) Evaluation Metrics

The short-term binary model's output is a binary classification. By averaging the precision score (AP) and F1 score the model is evaluated. The average precision score gives a precision-recall curve is to achieve each recall threshold, with the increase in recall from the previous threshold used as the

$$AP=\sum_{n}(R_n-R_{n-1})P_n$$

weight:

where Pn refers to the precision and Rn refers to the recall at the nth threshold. F1value helps us to acquire an accuracy test and is can be defined by the harmonic mean of precision and recall of the test.

a) Model Structure Tradeoffs

It is important to balance the precision and complexity to ensure the practical utility of made hardware predictor in lowpower embedded system. The feasibility constraints are considered to explore the impact the some of hyperparameters and layer structures on the model performance. The common Parameters include RNN type, model structure, number of hidden neurons, decimal digits, and number of time steps.The RNNs and LSTMs are found by based on the values os AP, F1, recall , runtime, and degree of prediction. recall score.

B. Long-Term Regression Model

For regression model, mean absolute error (MAE) is used to evaluate the accuracy, The MAE achieved by the predictor for 320ms in advance is 0.018. The highest accuracy achieved by existing prediction methods is 0.024 MAE, and the longest prediction step is 500ms [4], which improved by 25% and 36%.

CONCLUSION

Thus this paper proposed a Long Short Term Prediction Engine which was a new LTSM–based method for hardware hazard prediction. There are two models used by the prediction engine to provide a prediction of both normal conditions and urgent, which have different prediction requirements. In the ODROIDXU3 platform, the integrated model is trained and tested on data collected. Exact binary predictions are done by short term model in the critical conditions 40ms in advance and it reaches 0.78.

REFERENCES

[1] A. X. M. Chang, B. Martini, and E. Culurciello, "Recurrent neural networks hardware implementation on fpga," *arXiv preprint arXiv:1511.05552*, 2015.

[2] Z. Chen, Y. Liu, and S. Liu, "Mechanical state prediction based on lstm neural netwok," in *Chinese Control Conference*, 2017.

[3] A. Chigurupati, R. Thibaux, and N. Lassar, "Predicting hardware failure using machine learning," in *Reliability and Maintainability Symposium*, 2016.

[4] R. Cochran and S. Reda, "Consistent runtime thermal prediction and control through workload phase detection," in *ACM/IEEE Design Automation Conference*, 2010.

[5] A. K. Coskun, T. S. Rosing, and K. C. Gross, "Utilizing predictors for efficient thermal management multiprocessorsocs," *IEEETransactions on Computer-Aided Design of Integrated Circuits and Systems*, 2009.

[6] F.D. d. S. Lima, G. M. R. Amaral, L. G. d. M. Leite, J. P. P. Gomes, and J. d. C. Machado, "Predicting failures in hard drives with lstm networks,"in *Brazilian Conference on Intelligent Systems*, 2017.

[7] Y. Ge, Q. Qiu, and Q. Wu, "A multi-agent framework for thermal aware task migration in many-core systems," *IEEE Transactions on Very Large Scale Integration Systems*, 2012.

[8] I. Giurgiu, J. Szabo, D. Wiesmann, and J. Bird, "Predicting dram reliability in the field with machine learning," in *ACM/IFIP/USENIX Middleware Conference: Industrial Track*, 2017.

[9] Hardkernel, "ODROID-XU," Tech. Rep. [Online]. Available: http://www.hardkernel.com/main/main.php

[10] R. Kumar, S. Vijayakumar, and S. A. Ahamed, "A pragmatic approach to predict hardware failures in storage systems using mpp database and big data technologies," in *IEEE International Advance Computing Conference*, 2014.

[11] S. Liu, G. Liao, and Y. Ding, "Stock transaction prediction modeling and analysis based on lstm," in *IEEE Conference on Industrial Electronics and Applications*, 2018.

[12] T. Muck, S. Sarma, and N. Dutt, "Run-dmc: Runtime dynamic¨ heterogeneous multicore performance and power estimation for energy efficiency," in *International Conference on Hardware/Software Codesign and System Synthesis*, 2015.

[13] S. Huang, C. Fung, K. Wang, P. Pei, Z. Luan, and D. Qian, "Using recurrent neural networks toward black-box system anomaly prediction,"in *IEEE/ACM International Symposium on Quality of Service*, 2016.

[14] S. Sharifi, D. Krishnaswamy, and T. S. Rosing, "Prometheus: A proactive method for thermal management of heterogeneous mpsocs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013.

[15] G. Singla, G. Kaur, A. K. Unver, and U. Y. Ogras, "Predictive dynamic thermal and power management for heterogeneous mobile platforms,"in *Design, Automation Test in Europe Conference Exhibition*, 2015.

[16] X. Sun, K. Chakrabarty, R. Huang, Y. Chen, B. Zhao, H. Cao, Y. Han, X. Liang, and L. Jiang, "System-level hardware failure prediction using deep learning," in *ACM/IEEE Design Automation Conference*, 2019.