Microprocessors and Microsystems 72 (2020) 102907

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Pods – A novel intelligent energy efficient and dynamic frequency scalings for multi-core embedded architectures in an IoT environment

K. Tamilselvan [a,*], P. Thangaraj [b]

[a] Department of Electrical and Electronics Engineering, Bannari Amman Institute of Technology, India
[b] Department of Computer Science Engineering, Bannari Amman Institute of Technology, India

ARTICLE INFO

Article history:
Received 13 June 2019
Revised 6 October 2019
Accepted 10 October 2019
Available online 13 October 2019

Keywords:
Predictors
Dynamic frequency scaling
Thread mapping
Extreme machine – Learning
ARM/Cortex
BAT

ABSTRACT

In the Advent of the Internet of Things (IoT), embedded architecture takes an important dimension in terms of energy and accomplishment. The embedded system needs more and more intelligent algorithms for better performance and energy efficiency to fit into an IoT scenario. Moreover, with the existence of high-performance multi-core embedded architectures, achievements of energy efficiency remains in the dark side of the research. Several algorithms such as dynamic frequency scaling, thread mapping, starvation methodologies were proposed in embedded architectures for efficient usages of clock frequencies and these features were used as the energy saving modes in which the consumption of energy in the embedded architectures are being controlled. But these methods have several backlogs which permits the use of consumption in the embedded architectures. Considering the above features, this paper proposes a new methodology PODS(Predictors for Optimized Dynamic Scaling) which integrates a powerful machine learning algorithm for scaling the clock frequencies by the input workloads and allocation of the core depending based on the workload. The proposed framework PODS has different phases of working namely workload extraction, characterization, and optimization using BAT algorithms and prediction extreme Machine - Learning. The algorithm was tested on ARM/Cortex architectures (Raspberry Pi 3 Model B+), an evaluated algorithm using the IoMT benchmarks and various parameters that include energy consumption, accuracy of detection/prediction was determined and analyzed. It is found that the implementation of the proposed framework in the test is seen resulting between 35 and 40% reduction in the consumption of the power.

1. Introduction

Internet of things (IoT) which is an amalgamation of hardware and software it plays wide important role in day to day smarter life. IoT leads to the estimated connection of 9 billion devices currently it may reach up to of 24 billion devices by 2024 [1]. IoT provides an intelligent framework in which it collects sensory data from devices and transfers to the server /user using the internet. This network model is used for remote monitoring and control of the devices. The main feature of IoT is the timely delivery of the data from the user to a machine or vice -versa. The embedded processor performs an essential role in an IoT scenario, to this collection of data from the sensors to the servers and receive the data from the server for controlling the devices. Since these embedded processors are used in a continuous function, the demand for energy consumption increases day by day. High energy expenditure in embedded processors is one of the multi-faced problems scenario of an effective IoT scenario [2]. Input workloads are also increases day by day, producing some profound strength consumption and performance remaining to be a real challenge. Several researchers use changing frequency and voltage computing in embedded architectures which dynamically reduce regularity and voltage and thereby energy [3]. The several embedded architectures are equipped with a set of acceptable frequencies, including this user running method that selects the regularity performed by its need. However, the DVFS diminishes the appearance of the way. Hence the examinations use several heuristics for the choice of the best scaling portion under the objectives. But again the problem of fixing the scale factor without compromising on the energy and performance is the real nightmare in an IoT environment. Also fixing the lowest frequency does not always lower the energy consumption. Hence intelligent algorithms which can fix the best scaling factor with less energy consumption and high performance are to be designed and implemented in the embed-

* Corresponding author.
E-mail addresses: tamilselvank@bitsathy.ac.in (K. Tamilselvan), deantsd@bitsathy.ac.in (P. Thangaraj).

https://doi.org/10.1016/j.micpro.2019.102907
0141-9331/

ded processors. The paper proposes the integration of the machine learning algorithm for problem solving. The framework PODS (a predictor of optimized Dynamic Scaling) is proposed which consists of the three important parameters such as the workload extraction and characterization, optimization and implementation of the computer training algorithms for the fixing and prediction of the cores in accordance to the cores.

This paper presents the idea of using the BAT algorithm for the best workload extraction to fit into a machine learning algorithm for intelligent fixing of frequency scaling and core allocation. The paper also provide a description of the usage of the extreme machine - Learning which are characterized by the single feed forward layer, high performance, and high speed computation. The proposed framework is implemented in Raspberry Pi 3 Model B+ which consists of CortexA-7 Architectures. This algorithm is tested during the execution phase. To estimate this representation of the suggested algorithm, Accepted the IoMT benchmarks since it finds more suitable for an IoT environment. The researcher's experiments were carried out on Raspberry Pi using IoMT benchmarks and developed the threads which could be integrated with Raspbian OS (Linux kernel). The energy consumption of the test bench with the normal DFS mechanism is analyzed.

This Article has been organized as follows:

Section 1 Deals with the relevant works by more than one author. Section2 displays the working arrangement of the advanced skeleton with the preliminary descriptions of the workload extraction, bat algorithms, and extreme machine – Learning. The methodologies for fixing the best frequency scale factor and prediction of the cores using the extreme machine – Learning are described in the Section3 . The experimental setup, details about IoMT benchmarks are gained in Section 4. The evaluation parameters such as analyses of energy consumption, accuracy of prediction were discussed in Section 5. Which shows a comparison of accuracy with other machine learning algorithms? Section 6 provides a summary of the work, results and future scope in the section-VI.

2. Related works

Zhijian Lu depicts a proper review-control algorithm for dynamic voltage/recurrence scaling (DVS) in a compact multimedia system to spare power while keeping up an ideal playback rate. This calculation is comparative in unpredictability to the beforehand proposed change-point detection algorithm. In contrast, it works in a better way in maintaining constant throughput as well as is not rely on the hypothesis of an exponential circulation of the frame decoding rate. The proposed framework controller can keep the normal frame delay below 10% of the expected result than 90% of the time, though the change-point identification calculation maintained the standard frame delay with 10% of the objective out of 70% or lessen the implementing a same timing for workload [3].

Kihwan Choi presents a system known as "workloads decomposition" in the CPU workload its disintegrated in different sections: on-chips and off-chip. The on-chip workload highlights the CPU clock cycles that are needed to execute directions in the CPU while the off-chip workload catches the quantity of external memory access clock cycles that are required to perform outside memory exchanges. At the point when joined with a unique voltage and frequency scaling (DVFS) strategy to limit the vitality utilization. This outstanding burden decay technique brings about higher vitality investment funds. The workload breakdown method is performed at run time dependent on measurements revealed. [4]

Albert Mo Kim Cheng gives a new Dynamic Mode EDF scheduling algorithm with changing workloads mainly. With the help of RET as well as accumulated slack at run-time, any one of the frequency setting formats such as Single, Dual and three-mode shall be applied. It is important to notice that we can save energy only through a single combination of the speed-switching transition points, number of speeds/modes as well as the frequency scaling factor for every mode. The result of the experiment says that with the RET pattern, this Dynamic-Mode DVS algorithm touches the minimum 15% energy savings when compared to traditional two-mode DVS scheme on hard real-time systems. In addition, this experiment also gives importance to Energy Transition overhead or Speed-switching as well as employ the basic test of given algorithm. Deadlines are meted out at appropriate time with a low definition of voltage scaling strategy. Moreover, minimum 14% saving the energy consumption upon a non-DVS method is being observed [5].

Jurn-Gyu Park conducted a research on Mobile GPU Graphics workload characterization for DVFS Design with focused on energy efficiency and user experience on regular smart phone. Through this work, he developed new innovations which highlight particular stages of the graphics pipeline individually and compared the connections between different graphics workloads and energy results as well as the performance of various mobile graphics pipeline phases. Without wasting the user experience, this experimental work's result outlines the chances for high efficient CPU hardware components and memory for energy saving and integrated DVFS policies across the mobile GPU. In order to gain high energy efficiency without compromising in graphics performance for good user experience, the end results on Nexus 4 smart phone clearly showed that it is significant to exemplified graphics workloads and GPU hardware. This experiment will enhance high energy efficient DVFS algorithms for mobile graphics rendering at the times of modifying mobile GPU architectures [6].

Jungsoo Kim proposed a new kind of online dynamic voltage and frequency scaling (DVFS) system which considers computational workload, runtime distribution,stall time and program phase behavior. The problem of online DVFS is dealt with two ways which are program phase detection and interphases workload prediction. The program segment detection is used characterize the current instant belongs to particular program phase. Further, it is used to obtain the assumed workload according to the detected programming phase [7].

M.E. Salehi adopted an adjustable and capable update interval method for frequency management and dynamic voltage. In this process, the voltage and frequency of the method for the periodic workloads have been planned including the maintenance of soft real-time deadlines. Through introducing the idea of the effective deadline as well as considering the advantage of the feasible correlation between the subsequent workloads. In order to present the efficient level of the method, comparisons between adaptive, fixed interval and oracle DVFS systems have performed using actual workloads of the packet processing applications and MPEG2 decoder. The experimental results proved that the suggested adaptive interval DVFS technique can save more power with low frequency updates in comparison with the fixed interval DVFS systems [8].

Robert Schöne creates a new CPU pulse cover which molds its wavelength regulating functions in order to fulfill the showcase pattern of values with specific measurement in the Central Processing Unit. It reduces this capability indulging in recollecting bounden works. Apart from the benefits from processing, there is no requirement regarding more reproduction or compelling purposes. Every ruler is packed and loaded at runtime openly. In order to interrupt this kind of work display or to contributed loads, the supervisor should arrange package and upgrade for other running structures using diagnostics. The outcome of the test gives couple different x86 64 inspection procedures expose this common energy allotment is a original input load pack up which decreases 4 percentage or 2 as per the representation intense applicability.

Nonetheless, slightly the medium work time increases and the potential expenditure box shall be diminished upon the pair percentage comprised standard. Moreover, very good input loads will appear and bestow a bounded accent increase to 7 percent. Additionally, developments in the specific region are also considered. [9]

Anup Das has come up with an idea of adjustable method for Energy Minimization in Multimedia applications on a multi core system. Multinomial Logistic Regression-based workload differentiation technique is basic to the above-mentioned method. Furthermore Multinomial Logistic Regression-based workload identifies the receiving video with improbablility into a distinct classes. Each class is connected with a frequency value pre-determined with the help of exact training set which leads to less energy consumption. This approach is powered by a multicore system using Linux. The multimedia applications' experiments show that the suggested method lessens the energy consumption by minimum 20% without performance degradation [10].

Shoumik Maiti enabled a new framework which comprises extended rang (DVFS), Thread, core mapping and runtime core selection working under turbo-boost (TB), nominal and near-threshold computing conditions. This method controls the process variation profile information of every core along with dark-silicon constraints in chip multiprocessors (CMPs) to choose map applications, cores and figure out the optimal frequency and voltage operation points of every core lessen the energy through throughput constraints or maximize throughput under power constraints. The results of this experiment urges the necessity for extended range DVFS and taken into account of process variation information. The blueprint which supports extended range DVFS ends in 14.6% more throughput compared to a nominal mode framework with DVFS and 15% of energy savings. In addition, the process-variation awareness of the suggested method ends in 3.7% of energy savings and 11.9% increase in throughput when compared to the previous work which could not control process-variation [11].

A. Sanjeevi Kumar was very much involved in developing a case for DVS algorithms especially aimed at Interactive Graphics Games. The major contribution was to provide a framework which classifies the workload of game applications. This method tries to find out whether the workload classification framework is able to design concrete DVS algorithms as well as the way the algorithms may vary from the algorithms which are used in video decoding applications. In this research paper, the framework, which is proposed, envisages the processor cycle requirements of frames and to apply the prediction to increase the frequency and voltage of the processor. Neverthless, extra works should be carried out to efficiently manage the different tables and predict/figure the frame workloads. The application's workload is predicted and divided by previous history and each frame is implemented with processor without distorting the quality of decoding [12].

Yen-Lin Chen created a framework of lightweight learning-directed DVFS system which uses counter propagation networks to identify and characterize the work behavior as well as analyze the good frequency/voltage set ups for the system. For better performance, an intelligent adjustment mechanism is given to the users as per their requirements. The different results of the experiment between this algorithms and various competitive techniques are processed on the Intel PXA270 embedded platforms and NVIDIA JETSON Tegra K1 multicore platform. As per the results, the learning –directed DVFS framework can pinpoint the appropriate (CPU) frequency along with the runtime statistical information of the current program including the energy savings of 42%. This interrogation help users to obtain effectiveness in consumption of energy as well as performance by mentioning the characters of performance loss [13].

Kevin K. Chang has taken a holistic idea to comprehend and utilizing the reliability and latency characteristics of updated DRAM when the supply voltage is decreased below the level of nominal voltage mentioned as per DRAM regulations. The highlight of this work is that it identifies if supply voltage reduces below a certain level, and then it creates distorted errors in the result. This experimental identification shows that the small errors in the result shall be avoided by increasing the latency of three main DRAM operations such as activation, restoration, and precharge). This method is classified the different relationships between error locations, and supply voltage, DRAM temperature, stored data patterns and data retention. Based on the results, this work suggests a Novel DRAM energy reduction mechanism is known as Voltron. The main objective of Voltron is to make use of a performance model in order to identify that how far can it decreases the supply voltage with no errors as well as without crossing the limitation of a user-specified threshold meant for loss. The results convey that Voltron decreases the system energy consumption and average DRAM by 7.3% and 10.5% respectively. Simultaneously, it limits the average system performance loss up to 1.8% for various workloads of memory-intensive quad-core. It proves that Voltron efficiently perform well than the previous dynamic frequency and voltage scaling mechanisms for DRAM [14].

Nadja Peters provided the deep knowledge for the web browser workload on HMP platforms with the objective of maximum power savings through the observations. This work focuses on the function calls produced by the web browser and actual thread workloads. Direct information without any ambiguity is given for power management. As per the characterization, the power management techniques such as power gating, DVFS and thread allocation to CPU cores are applied. In addition, this work outlines the capabilities of power saving for web browsing in Android. The beginning results of the system prove that the updated Android power management has enough space for enhancement and necessary operating system entities such as the governor, power control unit and scheduler to collaborate each other to gain maximum power savings [15].

Jacob R. Lorc is dealt with algorithms for dynamically varying (Scaling) CPU voltage and speed for better energy consumption. Through this system, we can come to know how to change any scaling algorithm to maintain performance level but lessen the proposed energy consumption. The proposed approach is called PACE (Processor Acceleration to Conserve Energy). Due to the dependent of PACE on the probability distribution of the task's work requirement, these methods for calculating the distribution and evaluation are based on different real workloads. Additionally, this work shows us how to calculate the optimal schedule which modifies speed level with limited allocations. The PACE approach gives very low additional overhead and produces valid reductions in CPU energy consumption. Simulations done by real workloads depicts that it minimizes the energy consumption of the CPU of past published algorithms by up to 49.5%, with an average of 20.6%, without affection the quality of performance [16].

Anshul gandh proposed an analytical model to comprehend the effect of workload qualities on the effectiveness of scale-up and scale out in the cloud. The system scrutinizes the core scalability of parallel and sequential workloads. This model gives exact accuracy rather than the implementation results as well as very helpful in analyzing the optimal scaling for all workloads given. The results of the experiment shows that joining scale-up and scale-out could give important cost savings on pure scale-out and scale up. Moreover, this method provides proof that scale-up has higher quality than scale-out if the SLA is strict but scale-out has higher quality if the load is high. Significantly, this method proves that the relative ordering of various scaling options which are depended

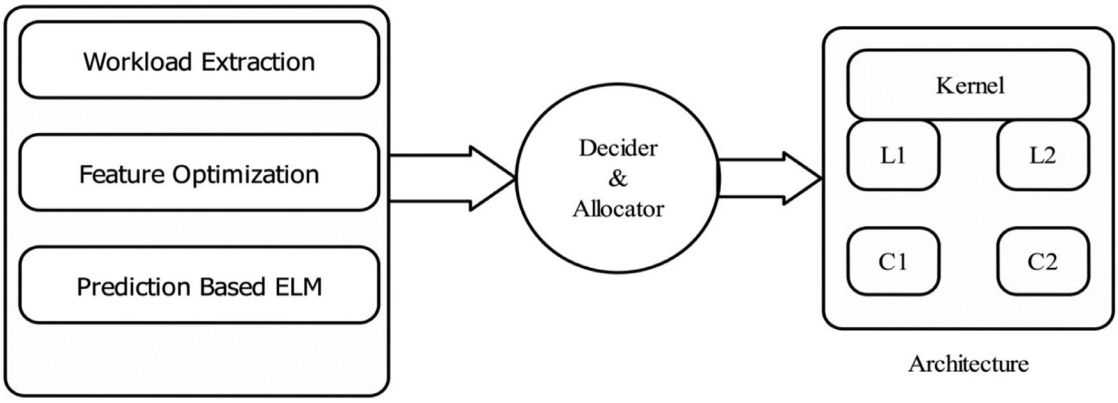

Fig. 1. shows the proposed framework. It consists of three stages which move detailed below.

on the workload characteristics. Vertical scaling is near-optimal for all loads and SLAs when the workloads have higher scale-up efficiency. Furthermore, the overall resources needed by the workload for gaining its SLA reduce according to the increasing ability in the scale-up efficiency [17].

2.1. Proposed algorithm

Fig 1 Shows the Overall Framework for the Proposed PODS Algorithm

As shown in Fig 1, the overall framework works on that three-tier mechanism which is detailed as follows as

2.2. Extraction phase

A workload is a collection of one or more programs. The first phase of the proposed framework deals with the workload extraction. Have listed the advantages, drawbacks of some of the existing tools which are used for the workload extraction.

2.2.1. Workload extraction tools-an overview

Wkldgen: wkldgen is a machine designed to decrease time to launch many importance as well as produce custom arrangement mechanics. The instrument itself allows flexibility in the following fields such as the determination of the type of statements, number of statements, number of threads, input dataset, and relationship arrangements.

PERF Tools: The performance of the Perf tools is used for analyzing the performance registers of the CPU. The major advantage of the PERF tool is lightweight, less overhead, flexibility for the usage of the different architectures. It can be integrated with any embedded OS kernel.

LIKIWid: This tool is used for analyzing the performance registers of the CPU. LikiWid is the same as that of Perf tools but it is targeted only for the Intel Xeon Architectures. Along with perf, Likiwid is used for analyzing the performance registers for the different architectures.

Considering the features of the above tools, PODS integrates the PERF tools for the extraction of the workloads which are used for inputs for analyzing. Each workload parameters which are extracted using PERF tools are tabulated in Table 1

Since the 12 workload parameters were extracted for the different inputs, optimizing the workloads to select the best feature is mandatory for fixing the best scale factor and to predict the cores following the input workloads.

2.3. Workload optimization

Several algorithms are used such as Genetic Algorithms (GA), and Particle Swarm Optimization, Ant Colony also used basic ways, and Firefly Algorithms, Are most popularly used for optimization. In section, we use the most popular algorithm called BAT Algorithms for the optimization of workloads which is extracted using perf tools.

2.3.1. Bat algorithm – an overview

The conventional bat algorithms is obtained based on their echolocation or bio sonar properties. Based on the repetition cancelation algorithms, XinShe Yang [18] (2010) developed the bats structure with the resulting three different idealized controls:

1 Each bat accepts the response location, view, and Position similarly 'recognize' the Xi. The data between food/Prey, conditions limits into any mysterious behavior.
2 Boat's run randomly. Positively among speed Vi about location xi in regularity fmin, modifying convolution term loud A0s to seek for Prey. All posterior can change specific wavelengths (or regularity) to the transmitted vibrations. Furthermore, these movements concerning oscillation discharged 2 [0,1], depends upon concurrence from the perfect targets.
3 Moreover, these loudness packaging interchanges toward actions, its believe one loudness changes of each generous (positive). A0 is an unchanging benefit Amin.

Separately bat Motion means connected including the quickness vit and initial distance xit with the 'n' number of iterations within each dimensional space approximately search space. Amongst group the bats, the best bat has to be chosen depends on the three rules which are stated above. The updated velocity vit and initial distance xit using the three rules is given below

$$\text{fi} = \text{fmin} + (\text{fmax} - \text{fmin})\beta \quad (1)$$

$$\text{xit} = \text{xit} - 1 + \text{vit} \quad (2)$$

where $\beta \in (0,1)$ fmin is the minimum frequency $= =0$ and fmax is the maximum frequency which initially depends on the problem statement. Each bat is initially allocated for the frequency between the fmin and fmax.

To get optimum solutions, Fig.2 it is essential for the variation of the loudnes including individual throb discharge. Since specific loudness customarily contracts bat should establish. while its rate of beat emission improvements, the loudness can be taken as either interpretation and monopoly. The engaged device of the BAT algorithm is represented in Fig.1

Table 1
Workload parameters extracted using the PERF tools.

Performance monitors	Description
Power [W]	The normal power dissipation in watts for a space-time collop
Time(s)	The period from the origin of achievement to end.
UNHALTED CLK CYCLES	The number of clock cycles that executed an instruction.
FP COMP OPS EXE X87	The amount of floating point instructions executed
BR INST RETIRED ALL BRANCHES	A representation of branching pathways preparations executed.
BR INST RETIRED CONDITIONAL	Several branches executed conditionally.
BR MISP RETIRED ALL BRANCHES	The number of mispredicted sprig predictions.
INSTR RETIRED ANY	The number of instructions executed of any type.
INT MISC STALL CYCLES	Stalls from something other than Load/Store operations
L3 LAT CACHE MISS	The number of L3 cache for an Execution misses.
ICACHE MISSES	The bulk of direction cache misses.
L2 RQSTS MISS	As L2 resources applications that occurred in a miss

Table 2
Specifications of BAT Algorithm used for workload optimization.

Sl. no	Title	Functions
01	No of BATS	04
02	Initial velocity	10%
03	No of iterations	50
04	Initial loudness	0.9
05	Initial throb rate	0.9
06	Least frequency	0 KHZ
07	Greatest frequency	20 KHZ
08	No of parameters obtained	04

The features such as power, cache size, branch prediction and time were extracted from 12 workload parameters using the above BAT algorithm.

Since we have 12 workload features, the following BAT specifications are tabulated in Table 2 which are used to optimize the workload features which are obtained from the PERF tools

2.4. Prediction using ELM

2.4.1. Extreme machine – Learning – a preliminary VIEW

After extracting the features, the proposed PODS uses Extreme machine – Learning proposed by G.B.Huang [19], in which each network practices the single separate layer, higher preparation speed, best accuracy/generalization, and universal function calculation capabilities[20,]. Network like this type has 'L' neurons in the hidden layer which are needed to operate with an infinitely differentiable activation function (for instance, the sigmoid separation), however much of the output course is in linear. The hidden layer should not be tuned mandatorily in the process of ELM. The hidden layer's weights are randomly given along with bias weights. It does not mean that hidden nodes are unnecessary but there is no need of tuning them. In addition, the hidden neurons parameters could be arbitrarily generated in the initial stage which means before managing the training set data. For a single-hidden layer ELM, the network result is provided by

$$f_L(x) = \sum_{i=1}^{L} 1\beta_i h_i(x) = h(x)\beta \tag{3}$$

where x represents the input and β is the output weight vector and it is given as follows as

$$\beta = [\beta_{21},\ \beta_{22},\ \ldots\ldots\ldots\ldots\beta_L]^T \tag{4}$$

H(x) is the output visual hidden layer which is represented by the eqn

$$h(x_2) = [h_1(x_2), h_2(x), \ldots\ldots\ldots\ldots..h_L(x) \tag{5}$$

To determine the Output vector O which is called as the target vector, the hidden layers are represented by Eq. (4)

$$H = \left[h(x_1)h(x_2) \vdots h(x_N)\right] \tag{6}$$

The basic implementation of the ELM uses the minimal non - linear least square methods which are represented in Eq. (5)

$$\beta' = H^*O = H^T(HH^T)^{-1}O \tag{7}$$

Here H∗ is the special form of the inverse of H called as Moore−Penrose generalized inverse.

Above eqn can be alternatively represented as follows

$$\beta' = H^T(\frac{1}{C}HH^T)^{-1}O \tag{8}$$

Hence the output function can be determined by using the above eqn

$$f_L(x) = h_{21}(x)\beta = h(x)\ H^T(\frac{1}{C}HH^T)^{-1}O \tag{9}$$

ELM uses the kernel function to produce the outputs with good accuracy. The major advantages of the ELM are minimal training errors and better approximation. Since ELM uses the auto-tuning of the weight biases and non-zero activation functions, ELM finds its applications in classification and prediction values. The detailed description of ELM 's equations can be found in [17,18], The pseudo code for ELM shown in Algorithm rule 1

Step 1: Training Sets of 'N' data with an Activation Function and n Hidden neurons
Step 2: Input weights and biases were assigned.
Step 3: Calculate the hidden matrix H
Step 4: Calculate the Output weight Matrix β
Step 5: Classify /Predict the values

2.4.2. ELM for the fixing and prediction

Table 3 shows the After fixing the best scale factors, ELM is used for the prediction of the cores following the input workloads. the rule sets which are used for the prediction of the cores following the inputs are described in Table 4

Pt, Tt, CMRt, CHRt are considered to be thresholds used in the PODS which are decided based on thumb rule and it varies which depends on the embedded architectures.

The experimental setup, dataset descriptions were detailed in this section.

3. Setup

The proposed PODS algorithm is tested on Raspberry Pi 3 Model B+ which consists of Cortex-A based quad core architecture which runs with the Frequency of 1400 MHZ with the Raspbian OS (Linux Based). The experimental setup is shown in Fig.3. The different frequency scaling such as F1 = =600 MHZ, F2 = =800 MHZ, F3 = =1200 MHZ, F = =1400 MHZ

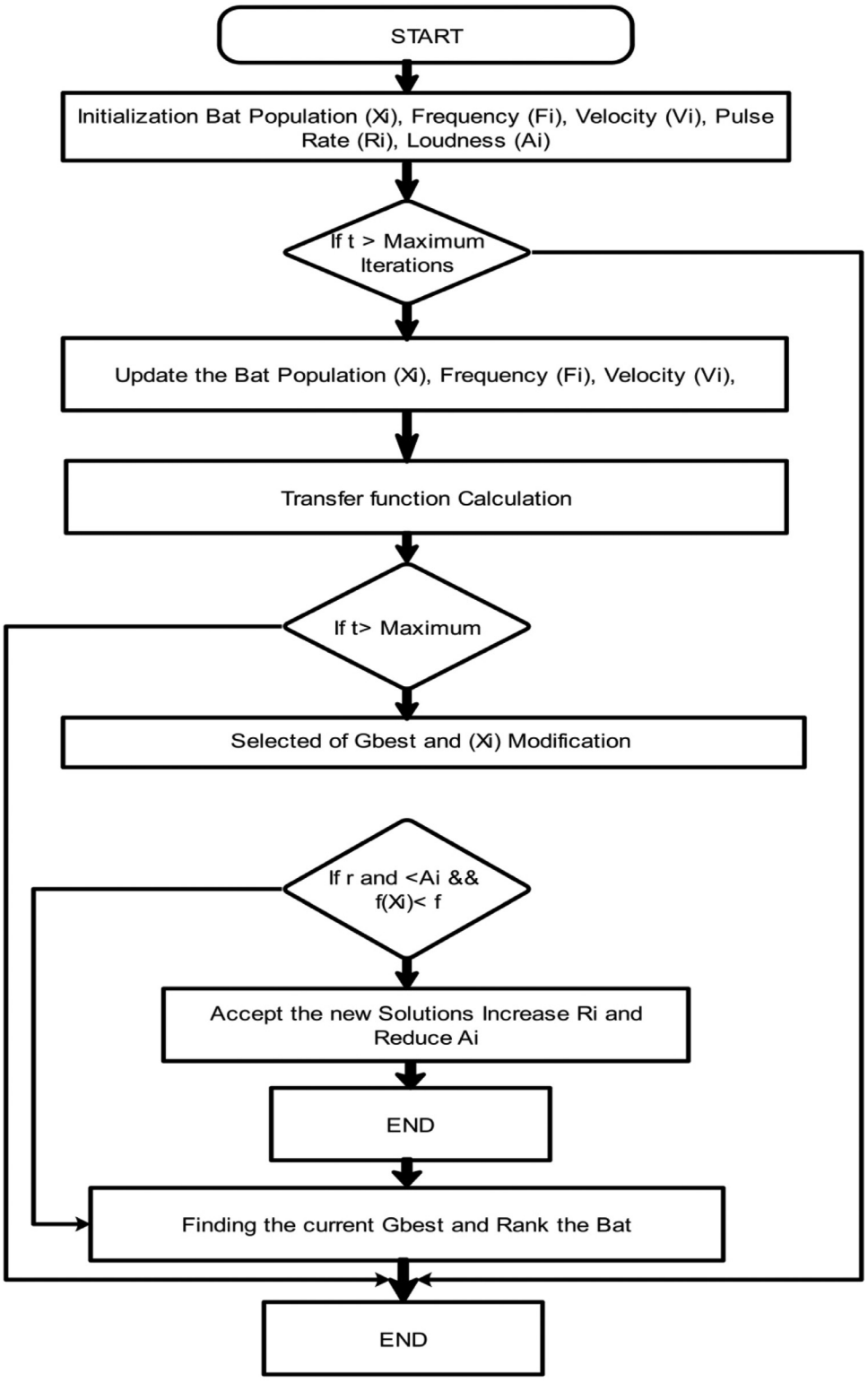

Fig. 2. Flow Diagram for the Working Mechanism of BAT Algorithms.

Table 3
ELM parameters used in the PODS.

Sl. no	Title	Functions
01	Network type	Multilayer networks
02	Activation variable	Sigmoid functions
03	No of hidden layers	150
04	Input parameters	Workload parameters (Power, Time, I Cache Hit ratio, Cache Miss Ratio)
05	Output parameters	Frequency scaling factors
06	Classification type	Multi-Classification (Scaling and Core Architecture)

Table 4
Illustration Of ELM Rule Sets for the Fixing the Scale Factor and Prediction of Core Architectures.

SI. no	Workload parameters				Output formation	Core architecture
	Power	Time	Cache Size Miss Ratio(CMR)	I Cache Hit Ratio(CHR)		
01	P==Pt	T==Tt	CMR=CMRt	CHR=CHRt	F1	C1
02	P>Pt	T>Tt	CMR>CMRt	CHR>CHRt	F2	C2
03	P<Pt	T<Tt	CMR<CMRt	CHR<CHRt	F3	C3

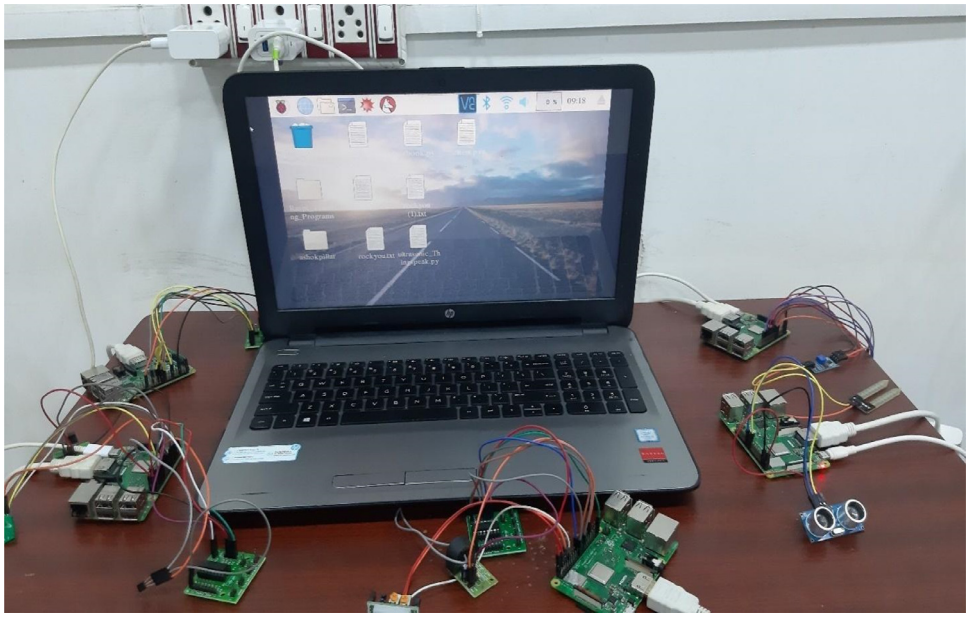

Fig. 3. Raspberry Pi 3 Model B+ testbed which is used for implementation of the proposed framework.

3.1. Dataset descriptions

Different workloads ARE obtained from Different sensors, such as cameras, image processing techniques are collected from the experimental setup as shown in Fig. 3. Nearly 1600 datasets were used for training the proposed PODS framework. For testing and evaluating the proposed algorithm, we have used IoMT benchmarks which spanning the variety of medical applications such as inverse randon transform, heart waveforms and variety of the image processing applications.

4. Results and discussion

4.1. Power consumption analysis

The power consumed is analyzed before and after implementation of the Proposed PODS framework in tested. The various analyses of the power w.r.t time are shown in preceding Fig. 4 (Fig. 1).

From the above Figs. 4. and 5. it shows the implementation of the proposed PODS algorithm has reduced the power consumption by 45% of the total consumption while running the IoMT benchmarks in Raspberry Pi 3 when test boards use only predicted core C1.

From the above Figs. 6. and 7. it shows the implementation of the proposed PODS algorithm has reduced the power consumption by 35% of the total consumption while running the IoMT benchmarks in Raspberry Pi 3 when test boards use predicted two cores C1 and C2.

From the above Figs. 8 and 9, it clearly shows the implementation of proposed PODS algorithm has reduced the power consumption by 35% of the total consumption while running the IoMT benchmarks in Raspberry Pi 3 when test boards use predicted three cores C1, C2, and C3.

From the above Figs. 10–12. it shows the implementation of the proposed PODS algorithm has reduced the power consumption by 30% of the total consumption while running the IoMT benchmarks in Raspberry Pi 3 when test boards use Four cores C1, C2 C3, C4.

After analyzing the power after the implementation of PODS algorithms, the overall fixing of the frequency following the workloads and core allocations are summarized in Table 5

Total frequency scaling and energy consumed are depicted in Table 5. The total energy consumed by the raspberry pi 3 is reduced by nearly 35–40% of the energy consumed during the normal DFS enabled applications.

4.2. Accuracy detection

For evaluation of the accuracy of proposed ELM, 70% of data were used for training and 30% of data were used for testing. The evaluation is carried out for the IoMT benchmarks with the following mathematical expressions.

$$\text{Accuracy} = \frac{\text{DR}}{\text{TNI}} \text{ x100} \tag{10}$$

where DR and TNI represents number of detected results and total number of iterations. Fig 12 shows the comparative analysis between the ELM and other machine learning algorithms such as Support vector machines (SVM), Random Forest(RF), Naïve Bayes(NB) and Artificial Neural Networks(ANN)which are implemented in Raspberry Pi 3 using Scikit-learn -Python Environment. Fig 13 clearly shows that the ELM used in the PODS framework outperforms the other learning algorithms and produces 99% of accuracy which finds an important role in the prediction and detection of the cores.

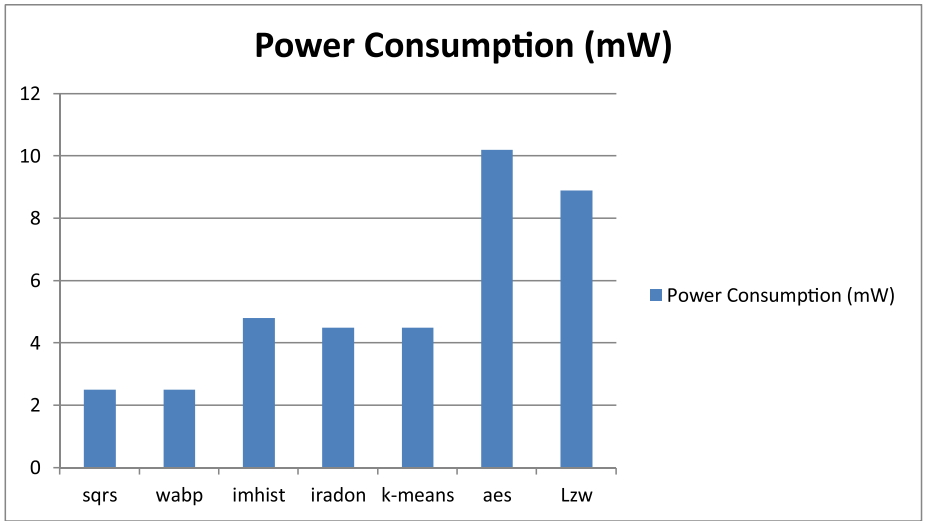

Fig. 4. Shows the power consumption of the IomT Benchmarks before PODS Implementation.

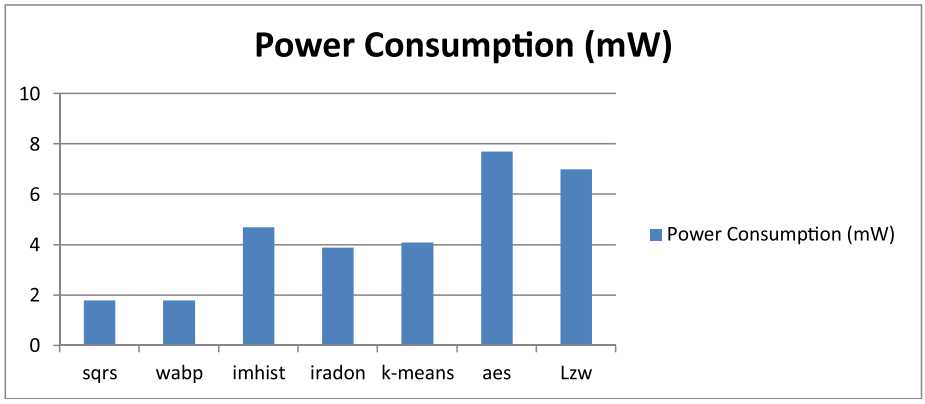

Fig. 5. Shows the Power Consumption of the IomT Benchmarks after PODS Implementation.

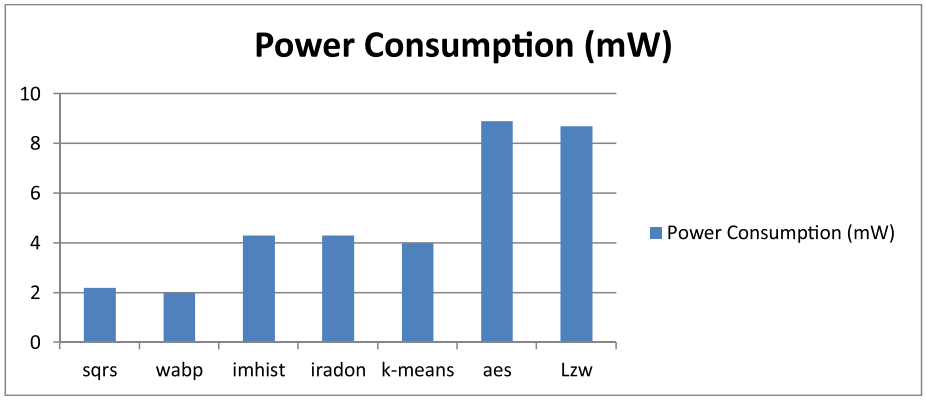

Fig. 6. shows the power consumption of the IomT benchmarks before PODS implementation.

Table 5
Frequency scaling factor used in raspberry Pi 3 test bed before and After the Implementation of PODS algorithms.

Sl. no	Details of features	Before PODS implementation	After PODS implementation	Energy consumed(%)
01	Frequency Used (3 IoMT benchmarks)- DFS Enabled	1400 MHZ	800 MHZ	60%
02	Frequency Used (4IoMT benchmarks)- DFS Enabled	1400 MHZ	800 MHZ	75%
03	Frequency Used (6IoMT benchmarks)-DFS Enabled	1400 MHZ	800 MHZ	100%
04	Cores Used(3 IoMT benchmarks)- DFS Enabled	02	01	25%
05	Cores Used(4 IoMT benchmarks)- DFS Enabled	02	02	30%
06	Cores Used(6IoMT benchmarks)-DFS Enabled	02	04	55%

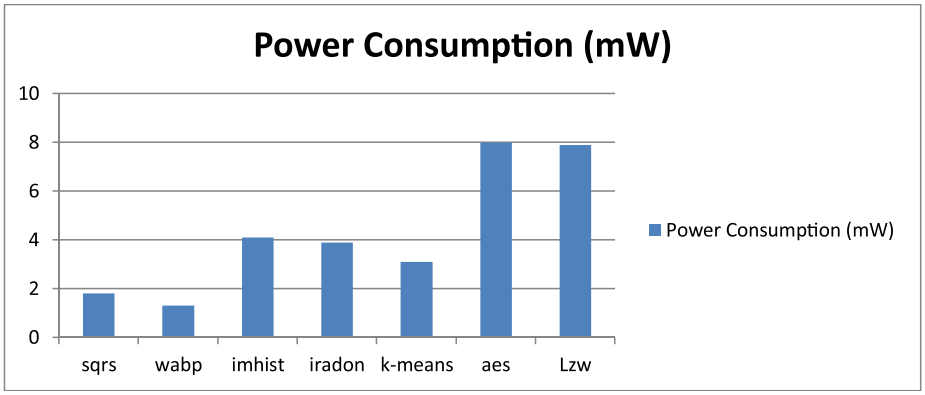

Fig. 7. shows the power consumption of the IomT Benchmarks before PODS implementation.

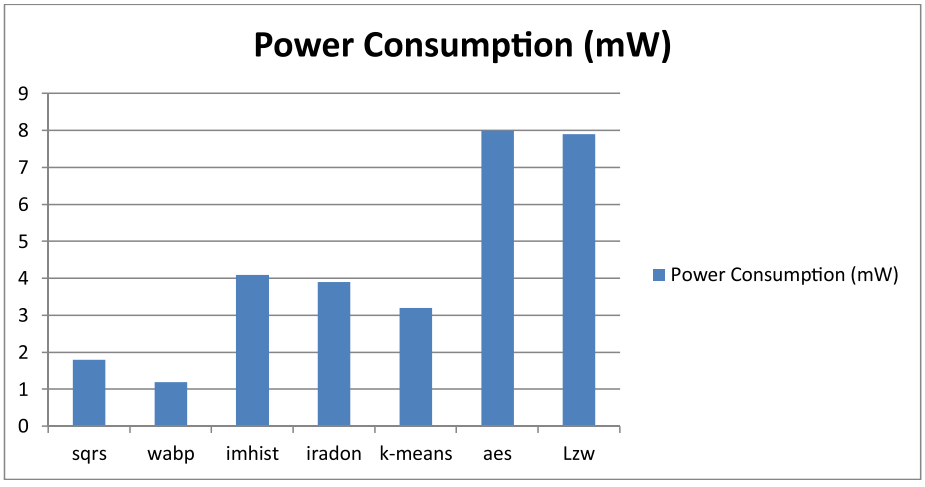

Fig. 8. Shows the power consumption of the IomT benchmarks before PODS implementation.

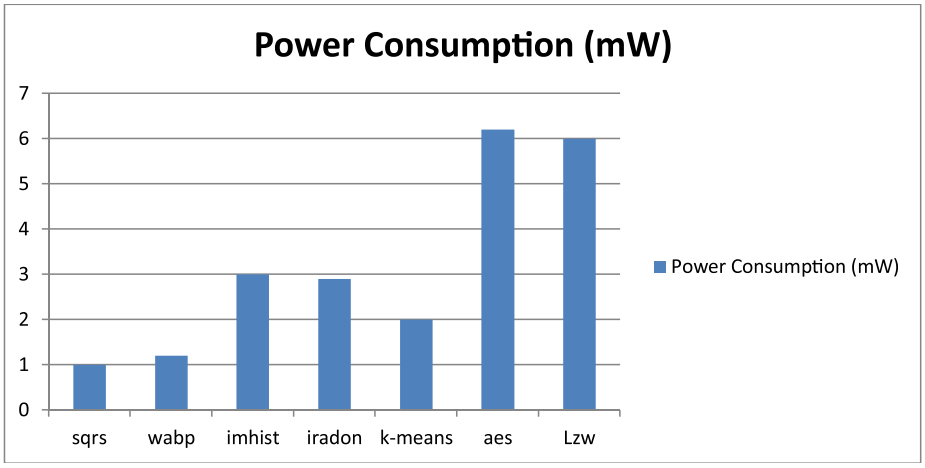

Fig. 9. shows the power consumption of the IomT Benchmarks before PODS Implementation.

5. Conclusion

The proposed framework PODS is designed and implemented in the multicore raspberry Pi architecture. The proposed framework is implemented as the Python API and tested during the execution of the programs. The power is analyzed for the different predicted cases and it was found that the energy consumption was reduced to 35–40% of the total energy consumed by the total cores . The implementation of the ELM frequency scaling methods in embedded system, proves to be vital and accuracy in predicting the cores is high as 99% when compared with the other existing algorithms. Hence the implementation of the intelligent algorithms in frequency scaling reduces the energy consumption which finds its way to work in an IoT environment. Further, these algorithms

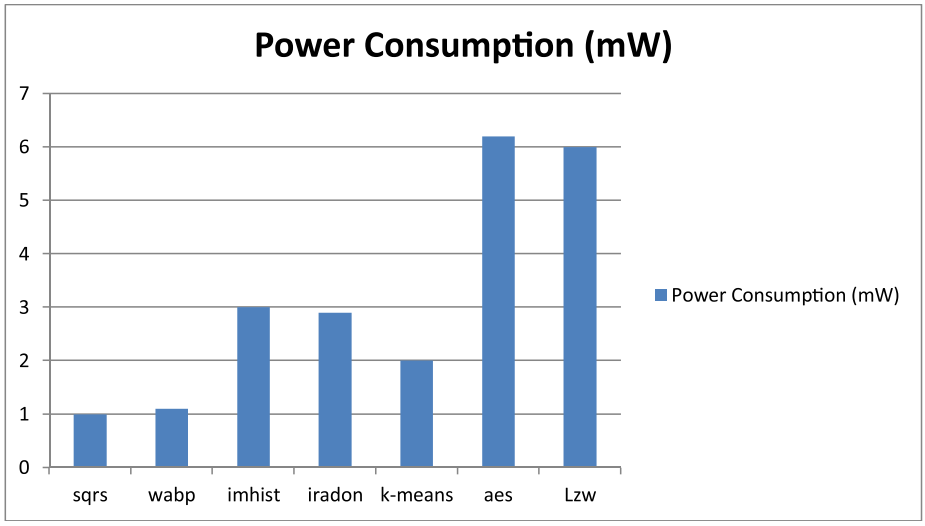

Fig. 10. shows the power consumption of the IomT Benchmarks before PODS implementation.

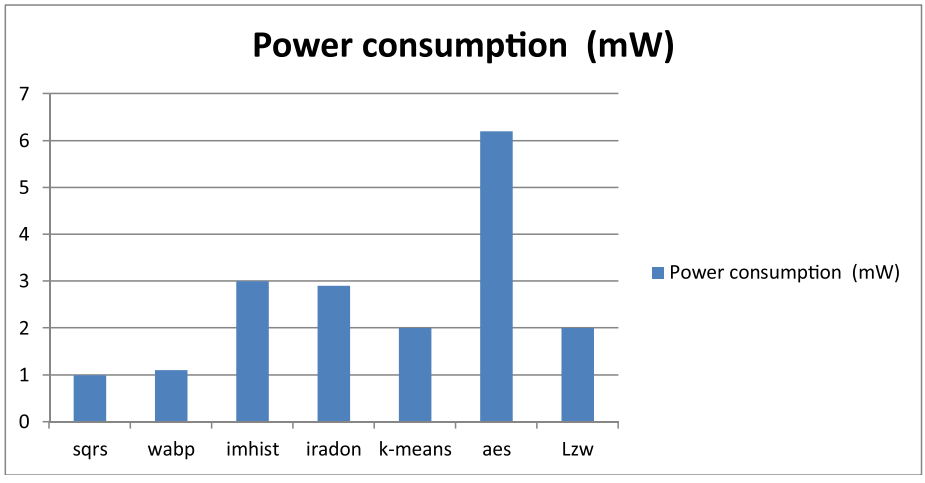

Fig. 11. shows the power consumption of the IomT Benchmarks before PODS Implementation.

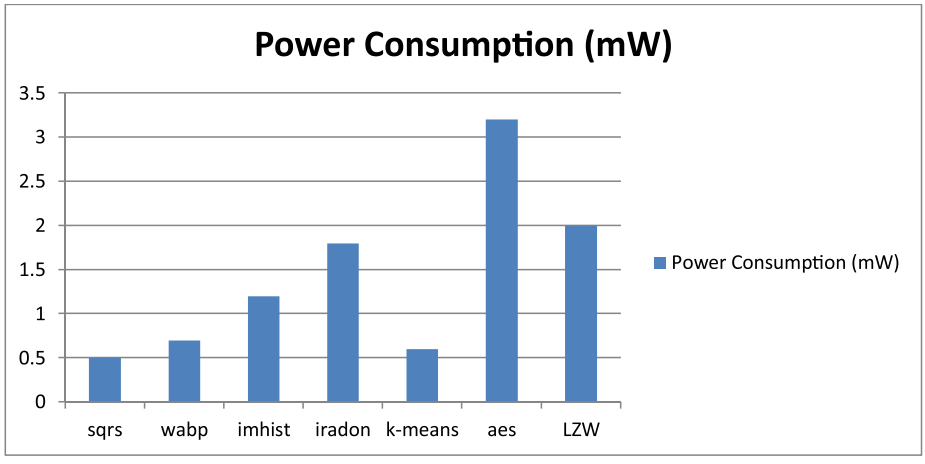

Fig. 12. shows the power consumption of the IomT Benchmarks before PODS implementation.

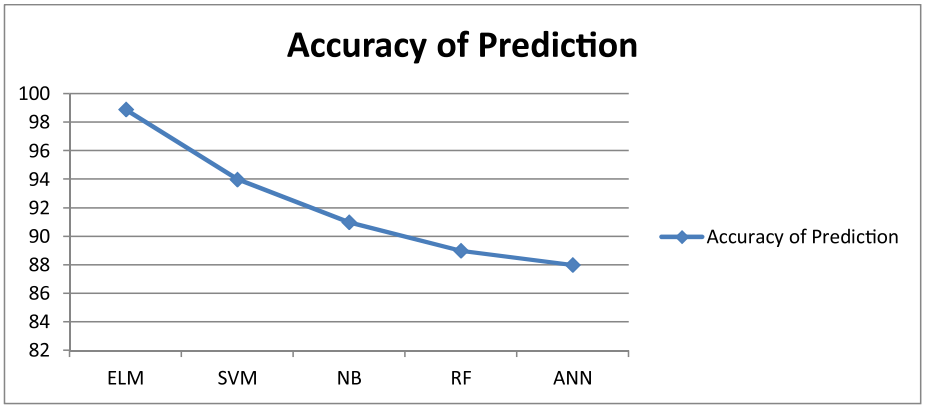

Fig. 13. Comparative analysis of accuracy of detection between the proposed algorithms and existing learning algorithm.

can be integrated with the kernel itself to schedule the workloads intelligently for better performances.

Declaration of Competing Interest

There is no conflict of Interest.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.micpro.2019.102907.

References

[1] P.P. Ray, A survey on Internet of Things architectures, J. King Saud Univ. – Comput. Inf. Sci. 30 (3) (July 2018) 291–319.
[2] V. Marinakis * ID, H. Doukas, An Advanced IoT-based system for intelligent energy management in buildings", Sensors 18 (2018) 610 ww.mdpi.com/journal/sensors, doi:10.3390/s18020610.
[3] Lu, Z. & Hein, J. & Humphrey, M. & Stan, M. & Lach, J. & Skadron, K.. (2002). Control-theoretic dynamic frequency and voltage scaling for multimedia workloads. 156. 10.1145/581652.581654
[4] K. Choi, R. Soma, M. Pedram, Dynamic voltage and frequency scaling based on workload decomposition, in: Proceedings of the 2004 International Symposium on Low Power Electronics and Design (IEEE Cat. No.04TH8758), 2004, pp. 174–179.
[5] A.M.K. Cheng, Y. Wang, A dynamic voltage scaling algorithm for dynamic workloads, J. VLSI Signal Process. 52 (1) (July 2008) 45–57.
[6] J.-G. Park, C.-Y. Hsieh, N. Dutt, S.-S. Lim, Quality-aware mobile graphics workload characterization for energy efficient DVFS design, in: IEEE 12th Symposium on Embedded Systems for Real-time Multimedia (ESTIMedia), 2014, pp. 70–79.
[7] J. Kim, S. Yoo, C.-M. Kyung, Program phase-aware dynamic voltage scaling under variable computational workload and memory stall environment, IEEE Trans. Comput.-Aided Des. Integr. Circt. Syst. 30 (1) (2011) 110–123.
[8] M.E. Salehi, M. Samadi, M. Najib, A. Afzali-Kusha, M. Pedram, S.M. Fakhraie, "Dynamic Voltage and Frequency Scheduling for Embedded Processors Considering Power and Timing Constraints", IEEE Transactions on VLSI Systems, pp. 1–5.
[9] R. Schöne, D. Hackenberg, On-line analysis of hardware performance events for workload characterization and processor frequency scaling decisions, ICPE'11, March 14–16, 2011.
[10] A. Das, A. Kumar, B. Veeravalli, R. Shafik, G. Merrett, B. Al-Hashimi, Workload uncertainty characterization and adaptive frequency scaling for energy minimization of embedded systems, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, pp. 43–48.
[11] S. Maiti, N. Kapadia, S. Pasricha, "Process variation aware dynamic power management in multicore systems with extended range voltage/frequency scaling,"
[12] A.S. Kumar, A. Rajalingam, B. Kokila, Frame-based prediction for games in mobile devices using dynamic voltage scaling, Int. J. Adv. Res. Comput. Commun. Eng. 1 (August (6)) (2012) 1021–2278.
[13] Y.-L. Chen, M.-F. Chang, C.-W. Yu, X.-Z. Chen, W.-Y. Liang, Learning-11. Directed dynamic voltage and frequency scaling scheme with adjustable performance for single-core and multi-core embedded and mobile systems, Sensors 18 (9) (2018) 3068, doi:10.3390/s18093068.
[14] K.K. Chang, et al., Understanding reduced-voltage operation in modern dram devices: experimental characterization, analysis, and mechanisms, Proc. ACM Meas. Anal.Comput.Syst. 1 (June (1)) (2017) 9 42 pages.
[15] Nadja Peters, S. Park, S. Chakraborty, B. Meurer, H. Payer, D. Clifford, "Web browser workload characterization for power management on HMP platforms," CODES/ISSS '16 October 01-07 2016, Pittsburgh, PA, USA, DOI: 10.1145/2968456.2968469.
[16] J.R. Lorch and A.J. Smit, "Improving dynamic voltage scaling algorithms with pace," ACM SIGMETRICS 2001 6/01 Cambridge, MA., USA
[17] A. Gandhi, P. Dube, A. Karve, A. Kochut, L. Zhang, Modeling the impact of workload on cloud resource scaling, in: IEEE 26th International Symposium on Computer Architecture and High-Performance Computing, 2014, pp. 310–317.
[18] X.-S. Yang, X. He, Bat algorithm: literature review and applications, Int. J. Bio-Inspired Comput. 5 (3) (2019) 141–149.
[19] G.-B. Huang, Q.-Y Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[20] B. Wang, S. Huang, J. Qiu, et al., Parallel online sequential extreme learning machine based on Mapreduce, Neurocomputing 149 (2015) 224–232.

Tamilselvan. K currently pursuing his Ph.D degree in Anna University, India. He received B.E Degree in Electronics and Communication Engineering From Anna University, India. He receives a Professional diploma in the embedded system from Open University Malaysia & M.E Degree at Anna University, Chennai. He is currently working as an Assistant Professor at Bannari Amman Institute of Technology, India. He has published more than 25 papers in journals & Conferences. His research interests include a Real-Time system & embedded system. Recently he has been awarded Best Faculty Award by Chief Minister of Pondicherry, India.

P. Thangaraj presently working as professor and Dean Academics & TSD in Computer Science and Engineering of Bannari Amman Institute of Technology, Sathyamangalam. He has completed his Ph.D. in Computer Networks from Techno Global University, Shillong. He also completed Ph.D. in Fuzzy Logic from Bharathiar University, Coimbatore. He did his M.E. (CSE) in V.M. University, Salem. He has 30 Years of teaching experience. Under his supervision so for 50 candidates completed their Ph.D in Anna University, Chennai. 5 candidates completed their Ph.D. in other Universities. He has published more than 120 papers in international journals. He has published two Books. He has received more than 2 crore of funds from different funding agencies such as AICTE, DST,UGC. Recently he has been awarded with ISTE—Anna University National Award for academic excellence.

Update

Microprocessors and Microsystems

Volume , Issue , , Page

DOI: https://doi.org/10.1016/j.micpro.2020.103505

Microprocessors and Microsystems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Corrigendum

Corrigendum to "Pods – A novel intelligent energy efficient and dynamic frequency scalings for multi-core embedded architectures in an IoT environment" [Microprocessors and Microsystems Volume 72, February 2020, 102907]

K. Tamilselvan [a,*], P. Thangaraj. P [b]

[a] Department of Electrical and Electronics Engineering, Bannari Amman Institute of Technology, India
[b] Department of Computer Science Engineering, Bannari Amman Institute of Technology, India

I regret to asking request for a corrigendum of the published paper titled "Pods – A novel intelligent energy efficient and dynamic frequency scalings for multi-core embedded architectures in an IoT environment" This paper is having Error on 2 nd Page. The Below image shows that Error.

Robert Schöne creates a new CPU pulse cover which molds its wavelength regulating functions in order to fulfill the showcase pattern of values with specific measurement in the Central Processing Unit. It reduces this capability indulging in recollecting bounden works. Apart from the benefits from processing, there is no requirement regarding more reproduction or compelling purposes. Every ruler is packed and loaded at runtime openly. In order to interrupt this kind of work display or to contributed loads, the supervisor should arrange package and upgrade for other running structures using diagnostics. The outcome of the test gives couple different x86 64 inspection procedures expose this common energy allotment is a original input load pack up which decreases 4 percentage or 2 as per the representation intense applicability.

Please remove it and use the following corrigendum text

Robert Schöne et al proposed a CPU frequency governor for handling the energy consumption efficiently among the computing resources. The proposed methodology takes into account the frequency of the processor based on the program counter measurements rather depending on the load. The developed methodology was also tested on two up-to-date x86_64 microarchitectures using SPEC OMP benchmark suites which proves that better performance is achieved in adaptive CPU frequency governors than the existing ones [9].

Nonetheless, slightly the medium work time increases and the potential expenditure box shall be diminished upon the pair percentage comprised standard. Moreover, very good input loads will appear and bestow a bounded accent increase to 7 percent. Additionally, developments in the specific region are also considered. [9]

DOI of original article: https://doi.org/10.1016/j.micpro.2019.102907.
* Corresponding author
E-mail address: k.tamilselvankesavan@gmail.com (K. Tamilselvan).

https://doi.org/10.1016/j.micpro.2020.103505