AI-Enabled Adaptive DVFS Controller Using LMS-Based Prediction and FSM Logic: FPGA and Real-Time MATLAB Implementation

Thomas Epsiba NJ[1], *Vimal* Rency A[2], *Latha* P[3], and *Samuel* Tensingh[4]

[1]Electronics and Communication Engineering, St. Joseph's College of Engineering, Chennai, India
[2]Electronics and Communication Engineering, St. Joseph's College of Engineering, Chennai, India
[3]Electronics and Communication Engineering, St. Joseph's College of Engineering, Chennai, India
[4]Digital Backend Engineer, Macquarie University, Research Assistant, Western Sydney University

Abstract. A popular method for maximising power and performance in contemporary processors and System-on-Chip (SoC) designs is dynamic voltage and frequency scaling, or DVFS. It is widely used in modern processors and System-on-Chip (SoC) designs to balance power and performance. However, conventional DVFS controllers often struggle with highly dynamic workloads, rapid temperature changes, and process uncertainties. Since they depend on fixed threshold values and predefined lookup tables, they tend to react slowly and can lead to unnecessary power consumption and lead to poor performance. So to avoid such limitations this paper proposes an AI-enabled adaptive DVFS controller using LMS-based prediction and FSM logic. At the same time, FSM states will ensure steady transitions between performance states and a real-time MATLAB environment is created to replicate the performance of FPGA-style AXI register behaviour, enabling cycle by cycle updation similar to hardware execution. This design is fully compatible with HDL coder, Vivado and AXI4- lite integration for FPGA implementation. Simulation results show that the predictor produces stable forecasts, the FSM responds smoothly with hysteresis, and the MATLAB real-time imitates closely resembles hardware operation. The proposed architecture offers intelligent and efficient DVFS control with very low computational overhead, making it practical for edge devices and power-sensitive SoCs.

1 Introduction

Power efficiency and performance are the central requirements of a modern embedded system, especially those running AI workloads that fluctuate unpredictably. DVFS is a commonly adopted technique to manage dynamic power by adjusting voltage and frequency based system activities.

The traditional DVFS methods still face several limitations such that they rely on fixed utilization thresholds, predefined lookup tables and temperature variations. In practical scenarios it gives responses with delays, rapidly changing workloads cause oscillation, unnecessary state switching, delayed frequency adjustments and high power consumptions. Additionally, variations in temperature and manufacturing processes further complicate the behaviour of static DVFS mechanisms.

To overcome these issues ,there is a growing interest in integration of intelligence into DVFS decision making. In this work, we introduce an adaptive DVFS controller that combines a ML model with a finite state machine(FSM) .

Thereby with many more ML based prediction based approaches such as linear regression and reinforcement learning where these approaches are too heavy for on-chip deployment, introduce high latency, and require more hardware resources. So, this paper proposes least mean square (LMS) method as it is Light-weight, Hardware-friendly, Easy to update in real-time, Capable of online learning.

The LSM- based prediction estimates the upcoming workload treads, enabling the system to anticipate changes rather than react to them. By blending the prediction and structured control, the proposed approach provides faster response, improved power efficiency, and better adaptability under dynamically changing operating conditions.

2 Literature Survey

The design of the DVFS controller begins with understanding the limitations in classical power management procedures and making a note of recent advancements in predictive-learning-based DVFS controllers. Traditional techniques depend on reactive thresholds based on temperature or CPU workloads. This often has poor performance and can lead to oscillatory workload changes. Recent predictive-learning-based DVFS such as - RL -based DVFO frameworks[2], deep learning workload predictions [14] and learning-discreted DVFS controllers for edge systems[3],[11] - demonstrate energy gains and reveal practical hurdles. These methods require large memory, complex algorithms making them unsuitable for hardware deployment. Lightweight workload predictors such as the LMS based model provide low computational complexity , as demonstrated in on-chip DVFS experiments [5],[27] . Likely, FPGA and SoC-based DVFS systems [4], [6], [18], [26] emphasize the need for low latency control logic that responds within a few clock cycles.

DVFS on FPGA and SoC has been done through voltage scaling frameworks, dynamic reconfiguration and processor-side frequency control [4], [6], [12]. These studies demonstrate the practical adaptation of DVFS for multimedia and AI workloads but lack an on-chip predictor or synthesizable controller. Thermal-aware DVFS has been applied on cloud and aerospace systems to manage energy budgets in harsh constraints [15], [19]. From these literature, following are evident:

1. ML-based DVFS approaches are too complex for on-chip implementation;
2. Few works describe complete synthesizable DVFS controllers with hysteresis and states.
3. Lightweight predictors like LMS are underutilized despite being most suitable for hardware DVFS

Thus above said features motivate the development of **simple, hardware-friendly DVFS controlle**r that combines lightweight prediction with FSM control.

3 Proposed Methodology

The proposed AI-enabled Adaptive DVFS system combines a lightweight LMS based predictor with a Finite State Machine (FSM) to achieve real time , predictive power management. This methodology focuses more on accurate workload prediction, stable performance-state transitions, and hardware-friendly implementation suitable for FPGA or SoC platforms.

3.1 System Architecture

The system is composed of four main components that are as follows

1. **Workload input :** The workload signal, normalized between 0 and 1, representing the instantaneous computational demand on the processor.
2. **LMS predictor :** the Least Mean Square (LMS) predictor estimates the next workload value based on the historical data or the trained data.The prediction formula states that :
   $y[n] = W \, . \, prev + bias$
   $W[n+1] = W[n] + \mu \, . \, (u\,[n] - y\,[n]) \, . \, prev$
   The $y[n]$ drives three-state DVFS finite-state machines (FSM) **,** namely Low , Medium and High performance states.
   Advantages: lightweight computation, fast convergence, and online adaptability.
3. **Finite State Machine (FSM) Controller:** The FSM works in implementing three DVFS performance states such that
   0 → Low, 1 → Medium, 2 → High
   This uses predicted workload to determine the state transitions. Incorporates Hysteresis thresholds to prevent rapid toggling :
   $\mathbf{T_1 = 0.08, T_2 = 0.26, H = \pm 0.02}$
   This implements minimum state residency (2 cycle) and transition latency (1 cycle) to ensure smooth and stable operation.
4. **Real-Time Simulation :**
   To imitate FPGA execution, MATLAB registers are used to emulate AXI4-lite interfaces :
   - u_reg : stores input workload
   - pred_reg: holds LMS-predicted workload
   - p_state_reg : stores the current DVFS state

   The real time loop updates all the modules cycle by cycle by replicating hardware timing. This provides a direct path to hardware development using HDL coder and vivado

3.2 Implementation

The DVFS controller is implemented as a sequential flow that executes once every sampling period(0.1). At every sample time , the workload is read, a prediction is computed and state is updated.

1. **Read workload input :** At each sampling time k , the current value $u(k)$ is read from input sources (test pattern or external interface).
2. **Predict workload :** The LMS predictor uses previous input and internal weight to estimate workload:
   a. Compute the predicted value $\hat{u}[k]$ from previous input
   b. Also clamp the predicted value into a range of 0-1
3. **Compare predictions with thresholds:** The LMS predictor value $\hat{u}[k]$ is compared with 2 thresholds:
   a. Lower threshold region for low performance state
   b. High threshold region for high performance states.
   A small hysteresis band is used to avoid frequent toggling.
4. **Determine DVFS state :** Based on the comparison :
   a. If prediction is low → **state = LOW**
   b. If prediction is medium → **state = MEDIUM**
   c. If prediction is high → **state = HIGH**
5. **Apply residency and transition :**
   Before actual change in state :
   a. Check whether the current state lasts at least for a minimum number of cycles.
   b. Check whether any transition delay has ended.
   If the conditions are met, the current state will be updated.
6. **Update outputs :** Once the state is confirmed:
   a. The DVFS state output is updated (0 = LOW, 1 = MEDIUM, 2 = HIGH)
   b. The workload, prediction, state is noted for

analysis The process continues for the next sampling instant and repeats in the same sequence for given workload input.

7. **Real - time execution(MATLAB) :** As far as real time simulation is concerned, in MATLAB the FPGA's behaviour is mimicked.
   a. A fixed time loop (Frequency= 100 Hz) hardware clock is simulated.
   b. AXI- like reading and writing is done through the "Registers".
   c. For each tick, the LMS predictor and FSM blocks are executed in the hardware.
   d. Live graphical datas is displayed for workload, prediction and state decision with transition in real time.
8. **Real - time execution(FPGA) :**
   a. Also using MATLAB - SIMULINK, the subsystem(LMS predictor + FSM) is converted into IP using HDL CODER .
   b. This IP is then exported to XILINX VIVADO . Here the custom IP is integrated with AXI
   - lite interface and Zync Processing Systems
   c. This System of connections is validated and bitstream is generated.
   d. Thus the generated bitstream is dumped in the FPGA and the real time transition for the input workload is seen.

4 Result and Discussion

The DVFS controller is evaluated using realtime MATLAB simulation and also FPGA implementation.

1. **Input workloads:** The LMS-predictor is given inputs that have varying conditions Datasets Three datasets namely trace_bursty.csv ,trace_heavy.csv and trace_steady.csv are created. The workload variation is given in below graph:

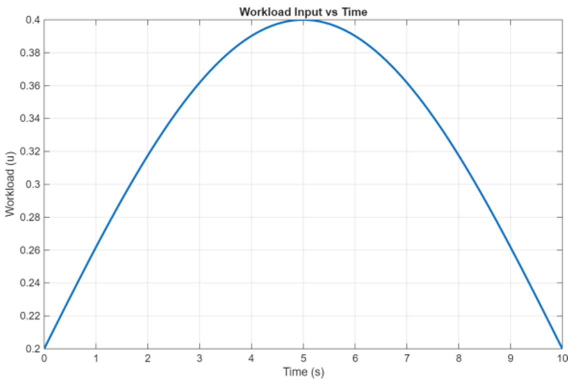

Fig. 1. Workload input vs Time

2. **Simulink Blocks Overview:** A simplified simulink model is developed for implementing and evaluating DVFS controllers before HDL generation. This includes:
   a. **WorkloadSource Block(From workspace):** This provides time varying workload as input .
   b. **LMS Predictor Block:** This is a MATLAB Function that computes the prediction value.
   c. **FSM Block:** This is also a MATLAB Function that determines the state of the input workload.
   d. **Power and Frequency mapping Blocks:** These blocks are used to equate the respective power and frequency levels for the given state.
   e. **To Workspace blocks :** This block is used for visualisation purposes. This can be attached to the output side of above blocks to visualize the output of particular blocks.

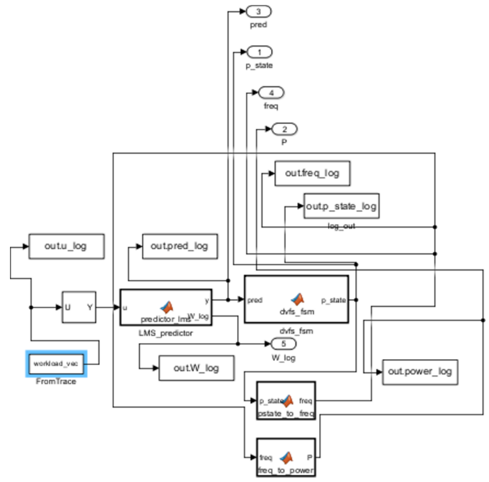

Fig. 2. Simulink block diagram

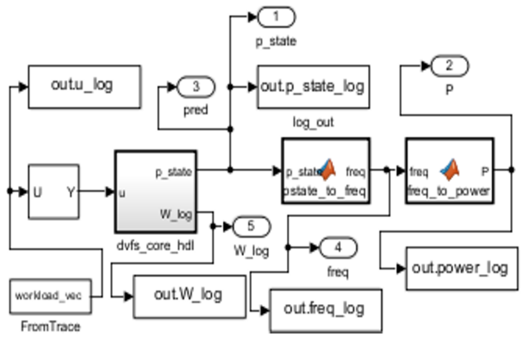

Fig. 3. Simulink block diagram with subsystem (LMS Predictor + FSM)

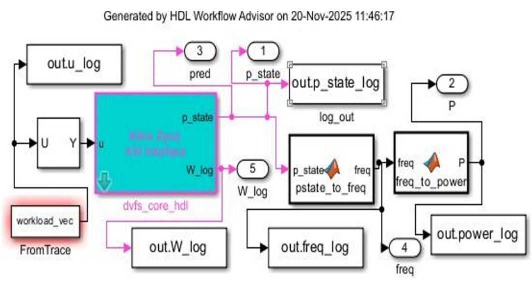

Fig. 4. Simulink block diagram with subsystem (LMS Predictor + FSM) , XILINX Zync and AXI Interface (from HDL Coder)

3. **LMS- Predictor Output:** The following shows how the incoming workload is tracked by predictor.Key observations:
   a. The predictor workload follows the actual workload without any sudden jumps.(Fig. 5)
   b. The curve rises and falls aligns with workload.
   c. The predictor stays within

realistic bounds (0.2 to 0.38). (Fig. 6)

d. The **RMS error** is **0.0218** and **accuracy** is **97.82%** which confirms high quality.

e. Fig. 7 indicates that there is no oscillation or divergence and the weight gets stabilized once error becomes small.

f. Also indicated LMS learning is **functioning correctly** and it is converging.

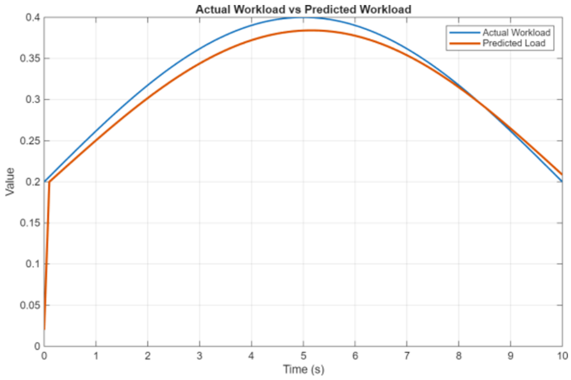

Fig. 5. Actual vs Predicted Workload

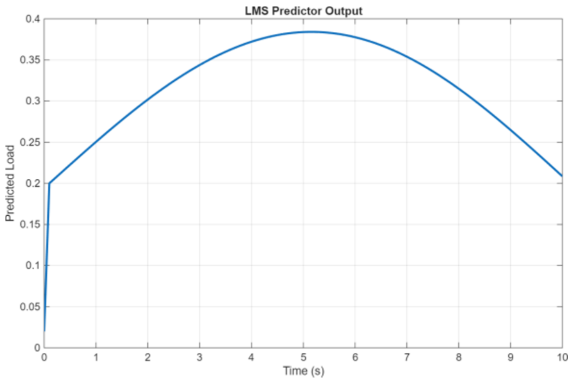

Fig. 6. LMS Predictor Output

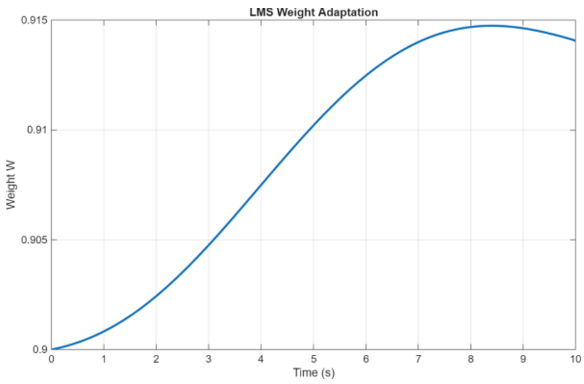

Fig. 7. LMS Weight Adaptation

4. **DVFS controller state transition:** The controller switches from state 0(low), state 1(medium) and state 3(high). The hysteresis and minimum-residency rule prevents unnecessary switching.

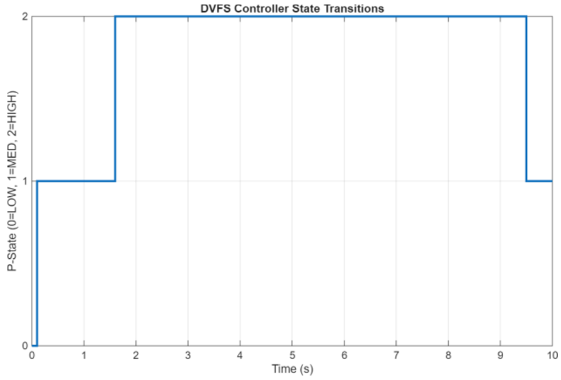

Fig. 8. DVFS controller state transition

5. **Frequency over time :** Frequency increases with high workload and decreases with low workload.

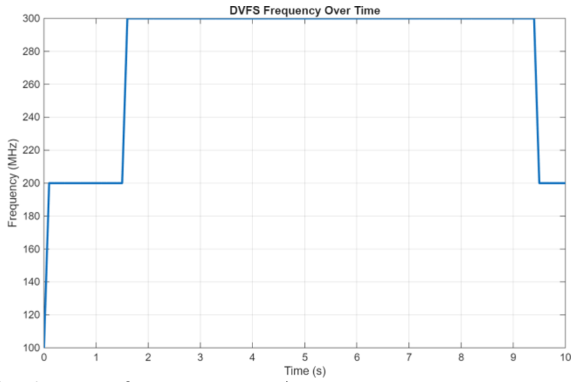

Fig. 9. DVFS frequency over time

6. **Power consumption over time :** Power consumption rises with the increase in the frequency and drops during low frequency periods there won't be any spikes or glitches in the output.

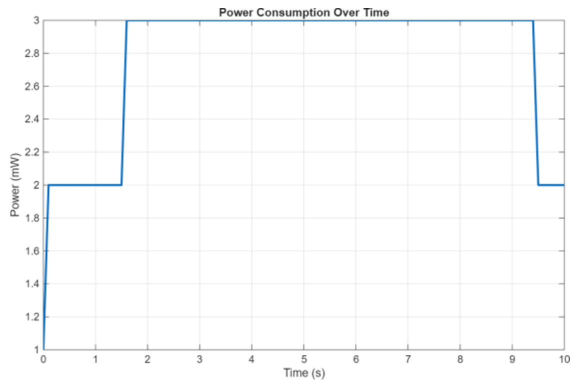

Fig. 10. DVFS power consumption over time

7. **Real time output:** When a real time FPGA environment is mimicked in MATLAB the output obtained . The observations are :

   a. The workload varies with time ranging from low to high intensity in a realistic pattern.

   b. The state of DVFS toggles between 0,1 and 2.

   c. The predictor value also changes

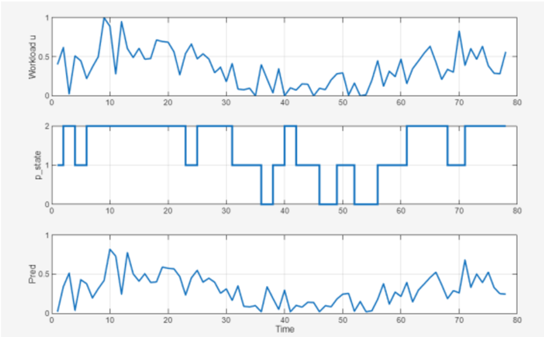

Fig. 11. Real time output from MATLAB

8. **Quantitative Analysis :** The DVFS - LMS predictive model demonstrates high accuracy and stable power performance behaviour.
   a. The LMS predictor has **very low RMSE(0.0218)**
   b. Prediction range matches workload pattern, demonstrating correct model learning.
   c. Running in Fixed mode(baseline power) consumes **3mW**. But when DVFS is taken into concern the adaptive power is **2.7 mW.**
   d. Thus , there is significant saving in power around **7.59%** .

```
===== DVFS + LMS MODEL METRICS =====
Min Workload          : 0.2000
Max Workload          : 0.4000
Min Predicted Load    : 0.0200
Max Predicted Load    : 0.3841
States Used           : [0 1 2]
RMSE                  : 0.0218
Accuracy              : 0.9782
Baseline Power (mW)   : 3.0000
Adaptive Power (mW)   : 2.7723
Power Saving (%)      : 7.59%
State Transitions     : 2
```
Fig. 12. Quantitative analysis

Overall, the integration system provides accurate prediction, robust noise tolerance, smooth transition changes and acceptable energy reduction.

5 Conclusion

This work presents an AI-enabled adaptive DVFS controller that integrates an LSM - based predictor with a finite state machine(FSM) to achieve intelligent and energy efficient and power management . The LSM predictor estimates the short term workload trends by allowing the FSM to actively adjust the processor's performance state while maintaining residency constraints. A real-time MATLAB mimic was developed to imitate FPGA AXI -style behaviour, providing cycle- accurate updates and verifying the performance before hardware development .

As a result the proposed controller ensures smooth transitions between DVFS states, reduces unnecessary switching and closely matches the expected hardware behaviour.

This approach is suitable for power constrained embedded systems and modern SOCs by providing both improved energy efficiency and faster response to dynamic workloads.

Overall, the combination of predictive intelligence and structured FSM control provides a **scalable and practical solution** for next-generation energy-aware processors.

6 Future Scope

The proposed system lays a strong foundation for the further enhancement in predictive DVFS and energy efficiency SOCs design. The possible future scope includes :

1. **Multi-Core and Heterogeneous Systems:** The method can be scaled to handle multiple processing cores or different SoCs, coordinating DVFS decisions across various workloads.
2. **Enhance Prediction Model :** It explores a smarter prediction model that is lightweight neural network, and hybrid prediction to achieve high accuracy without heavy hardware cost.
3. **FPGA Implementation :** By deploying the complete system on FPGA to test the DVFS controller with real time sensor inputs to verify and validates its performance in practical condition
4. **Scalability for Future SoCs:** Gives a scalable design for the upcoming low power IOT device, wearable electronics and high performance processors where adaptive and efficient power management is crucial.
5. **Improved Noise Robustness:** Further enhancing the noise immunity by experimenting with better filtering techniques to handle rapid fluctuating workloads

Reference

1. T.J. Zidar, T. Matić, I. Aleksi, Ž. Hocenski, Dynamic Voltage and Frequency Scaling as a Method for Reducing Energy Consumption in Ultra-Low-Power Embedded Systems. Electronics. 13, 826 (2024). doi: 10.3390/electronics13050826.
2. Z. Zhang, Y. Zhao, H. Li, C. Lin, J. Liu, DVFO: Learning-Based DVFS for Energy-Efficient Edge-Cloud Collaborative Inference, ACM Conf. Proc. (DVFO Preprint), (2023), doi: 10.48550/arXiv.2306.01811.
3. Y. Gong et al., Lotus: Learning-Based Online Thermal and Latency Variation Management for Two-Stage Detectors on Edge Devices, Proc. ACM/IEEE DAC, 61, (2024), doi: 10.1145/3649329.3657310.
4. L. Kechiche, L. Touil, M. Jemai, B. Ouni, A Power-Aware Real-Time System for Multi-Video Treatment on FPGA with Dynamic Partial Reconfiguration and Voltage Scaling,Eng. Technol. Appl. Sci. Res., 12, 8996–9004 (2022).
5. B. Dietrich, S. Nunna, D. Goswami, S. Chakraborty, M. Gries, LMS-Based Low-Complexity Game Workload Prediction for DVFS, Proc. IEEE ICCD, 417–424 (2010).

6. P. Mantovani, D. Giri, I. Loi, L. Benini, G. Di Micheli, An FPGA-Based Infrastructure for Fine-Grained DVFS in SoCs, Proc. ACM/IEEE DAC, 1–6 (2016).
7. X. Mei, Q. Xu, Y. Cao, X. Chu, A Survey and Measurement Study of GPU DVFS on Energy Efficiency, Sustain. Comput.: Inform. Syst., 3, 1–13 (2017).
8. P. Panda, A. Tripathy, K. C. Bhuyan, Learning-Based Ultra-Low-Power Optimization for VLSI Architectures, J. VLSI Circuits Syst., **7**, 131–144 (2025). doi: 10.31838/JVCS/07.01.15.
9. R. Hebbar, D. N. H. Thanh, A. Varma, An Experimental Evaluation of Workload Driven DVFS, Proc. ACM/SPEC ICPE Companion, 95–100 (2021).
10. Y. Tan, P. Malani, Q. Qiu, Q. Wu, Workload Prediction and Dynamic Voltage Scaling for MPEG Decoding, Proc. ASP-DAC, 1–6 (2006).
11. G. Zhou, W. Tian, R. Buyya, R. Xue, L. Song, Deep Reinforcement Learning–Based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions, Artif. Intell. Rev., **57**, 124 (2024).
12. P. D. S. Manoj, S. P. Dinakarrao, and A. A. Abouzeid, SmartDPM: Machine Learning-Based Dynamic Power Management for Embedded Systems, J. Low Power Electron., **14**, 345–360 (2018).
13. S. Ilager, R. Buyya, Machine Learning in Energy and Thermal-Aware Resource Management of Cloud Data Centers: A Taxonomy and Future Directions, Int. Conf. Comput. Syst. Ind. Inf. Proc., (2023/2024 preprint).
14. S. Tarsa, Workload Prediction for Adaptive Power Scaling Using Deep Learning, M.S. thesis, Harvard Univ., Cambridge, MA, USA, (2014).
15. A. Liubimov, Optimization of Energy Consumption of the CubeSat On-board Computer Using DVFS, Aerospace Technic Technol., 89–98 (2024).