**Journal of Computing and Electronic Information Management**
ISSN: 2413-1660 | Vol. 20, No. 3, 2026

# Neural Methods for Adaptive Voltage and Frequency Scaling in Energy Aware Processors

**Yihan Wang [1, *] and Suchuan Xing [2]**

[1] University of Pennsylvania, United States
[2] Duke University, United States
*** Corresponding author:** wangyh1@alumni.upenn.edu

**Abstract:** Dynamic Voltage and Frequency Scaling (DVFS) represents one of the most effective runtime mechanisms for balancing computational performance against energy consumption in modern processors. Traditional rule-based and heuristic-driven DVFS governors struggle to adapt to increasingly heterogeneous and unpredictable workload patterns characteristic of contemporary computing environments, including mobile devices, data centers, and edge deployments. This review examines the growing body of research applying neural methods—including recurrent neural networks, convolutional neural networks, deep reinforcement learning (DRL), and federated learning—to the adaptive DVFS problem. We survey the evolution from classical prediction-based approaches to policy-learning frameworks capable of jointly optimizing energy efficiency and Quality of Service (QoS) constraints. Our analysis reveals that DRL-based approaches consistently achieve energy reductions of 25–35% over fixed-frequency baselines while maintaining QoS violation rates below 4%, outperforming both traditional governors and supervised prediction models. We conclude by identifying open research directions in multi-objective optimization, on-device continual learning, and heterogeneous processor support.

**Keywords:** Adaptive voltage and frequency scaling; Deep reinforcement learning; Energy-aware computing; Neural network workload prediction; Processor power management; On-chip intelligence.

## 1. Introduction

The energy efficiency of computing systems has become a defining constraint across the entire spectrum of modern hardware, from battery-powered wearables to hyperscale data center clusters. Dynamic Voltage and Frequency Scaling (DVFS), the technique of adjusting a processors supply voltage and clock frequency at runtime in response to workload demand, stands at the center of processor-level power management strategies [1]. By exploiting the cubic relationship between dynamic power consumption and operating voltage, even modest reductions in voltage can yield dramatic energy savings, provided that performance targets are met within acceptable bounds [2]. The appeal of DVFS lies in its hardware accessibility—virtually all contemporary processors expose software-controllable performance states, commonly referred to as P-states, through standardized interfaces maintained by the operating system (OS) [3].

Despite decades of research and widespread deployment, conventional DVFS governors continue to exhibit significant inefficiencies. Reactive heuristics such as the Linux ondemand and schedutil governors operate on simple utilization thresholds, responding to workload changes only after performance degradation has already occurred [4]. Predictive governors employing time-series statistics or autoregressive models improve responsiveness but remain brittle under workload heterogeneity and phase transitions, which are increasingly common in mobile applications and cloud microservices [5]. As the complexity of computational workloads grows, there is mounting evidence that hand-crafted control policies cannot capture the high-dimensional correlations between hardware performance counters, thermal state, memory subsystem behavior, and future power demand [6].

Machine learning (ML) methods offer a compelling alternative paradigm. Rather than encoding expert knowledge into fixed rules, ML-based DVFS controllers learn mappings from observable system state to optimal frequency-voltage decisions either offline from historical traces or online through interaction with the hardware environment [7]. The rapid advancement of artificial intelligence (AI) hardware and efficient inference engines has made it increasingly practical to deploy non-trivial neural models within the power management stack with latency overheads measured in microseconds [8]. Early demonstrations using shallow regression models established proof of concept, but the field has since progressed to deep neural network (DNN) architectures capable of modeling temporal dependencies, multi-core interference, and heterogeneous workload contexts [9].

The proliferation of Internet of Things (IoT) devices and edge computing platforms has further intensified interest in neural DVFS. These systems operate under tight energy budgets—often relying on batteries or energy harvesting—yet must sustain responsive performance for latency-sensitive inference tasks [10]. At the same time, cloud and high-performance computing (HPC) environments increasingly deploy neural DVFS as part of broader AI-driven infrastructure management frameworks, where per-server efficiency gains aggregate into substantial operational cost reductions [11]. The convergence of these trends has stimulated a diverse body of research spanning supervised learning, reinforcement learning (RL), transfer learning, and federated learning (FL), each addressing different aspects of the adaptive scaling problem. This review provides a comprehensive synthesis of neural methods applied to adaptive DVFS in energy-aware processors, tracing the development of the field from early workload prediction models through modern deep reinforcement learning (DRL)

policy networks, and assessing the state of practical deployment.

## 2. Literature Review

The history of DVFS control spans over three decades, beginning with voltage scaling techniques introduced in early mobile processors and formalized through the Advanced Configuration and Power Interface (ACPI) specification [12]. Early academic proposals modeled processor workloads as cyclostationary processes, using linear predictors to anticipate future central processing unit (CPU) utilization and select appropriate P-states ahead of demand [13]. These interval-based prediction schemes demonstrated that proactive scaling could reduce energy consumption by 15–20% compared to purely reactive governors, establishing workload predictability as the central challenge in DVFS optimization [14].

The limitations of linear models became apparent as multi-threaded and heterogeneous workloads rose to prominence. Researchers turned to signal processing techniques including fast Fourier transforms and wavelet decomposition to identify periodic phase patterns in program execution, enabling more accurate frequency predictions across longer horizons [15]. However, these spectral methods required hand-tuned parameters and failed to generalize across application classes, motivating the adoption of data-driven approaches. Decision tree classifiers and support vector machines were among the first ML algorithms applied to DVFS, offering improved accuracy over linear regressors for workload phase classification [16]. These early ML governors operated on features derived from hardware performance monitoring units (PMUs), including instructions per cycle (IPC), last-level cache miss rates, and branch misprediction frequencies, which remain the dominant feature sources in subsequent work [17].

The deep learning era brought recurrent neural networks (RNNs) and, specifically, long short-term memory (LSTM) networks to the forefront of DVFS research. LSTM architectures are naturally suited to processor workload prediction because they can model long-range temporal dependencies in performance counter sequences, capturing phase transitions and bursty execution patterns that elude Markovian models [18]. Several studies demonstrated that LSTM-based predictors trained on per-benchmark traces could reduce prediction error for future IPC by 30–45% relative to autoregressive baselines, translating into meaningful energy-delay product (EDP) improvements when integrated into frequency scaling loops [19]. Convolutional neural networks (CNNs) were subsequently applied to workload characterization, treating performance counter time series as one-dimensional signals amenable to local pattern extraction via sliding convolution filters [20]. The combination of CNN feature extractors with LSTM temporal encoders yielded hybrid architectures that achieved state-of-the-art prediction accuracy across server, mobile, and embedded benchmarks [21].

Parallel to supervised prediction work, researchers began framing DVFS as a sequential decision problem amenable to RL. The core insight is that an optimal DVFS policy must account not only for instantaneous workload state but also for the downstream consequences of frequency decisions on temperature, performance service level agreements (SLAs), and future energy costs—a structure that maps naturally onto the Markov decision process (MDP) framework [22]. Early tabular Q-learning approaches demonstrated feasibility but were limited to coarsely discretized state and action spaces that could not capture the full complexity of real hardware [23]. The extension to deep Q-network (DQN) architectures enabled continuous or high-cardinality state spaces while retaining the convergence guarantees of temporal difference learning [24]. Subsequent work introduced actor-critic methods, including proximal policy optimization (PPO), which provided improved sample efficiency and stable training in non-stationary workload environments [25].

The thermal dimension of DVFS has received increasing attention as processor power densities have grown. Thermal-aware DVFS research acknowledges that exceeding the thermal design power (TDP) threshold triggers emergency throttling events that can violate latency guarantees and accelerate device aging [26]. Neural approaches to thermal-aware DVFS incorporate thermal sensor readings as additional state inputs, enabling policies that preemptively reduce frequency ahead of temperature-driven throttling rather than reacting after the fact [27]. Graph neural network (GNN) formulations have been proposed to model heat transfer between processor cores and memory subsystems on the same system-on-chip (SoC), capturing spatial thermal interactions that are invisible to per-core control loops [28].

The heterogeneous processor landscape presents additional challenges for neural DVFS. Modern SoCs feature multiple processor clusters with distinct frequency-voltage curves, including big-little core configurations in mobile processors and graphics processing unit (GPU) accelerators on server chips [29]. Learning a joint DVFS policy across heterogeneous compute domains requires multi-agent RL formulations or hierarchical policy architectures that decompose global energy optimization into coordinated per-domain scaling decisions [30]. QoS enforcement across heterogeneous domains further complicates reward design, as performance metrics differ fundamentally between latency-sensitive CPU tasks and throughput-oriented GPU workloads [31].

Transfer learning and domain adaptation have emerged as practical solutions to the data efficiency problem in neural DVFS deployment. Trained on large offline trace datasets, DVFS models can be fine-tuned on target hardware using a small number of online interaction samples, substantially reducing the time to reach production-quality performance [32]. FL approaches address the privacy and data sovereignty constraints that prevent centralized collection of workload traces from deployed devices, training DVFS models collaboratively across device fleets while keeping raw telemetry local [33]. Recent survey work has highlighted the need for standardized evaluation benchmarks and reproducible experimental protocols in neural DVFS research, with inconsistencies in workload selection, hardware platforms, and energy measurement methodology making cross-study comparisons unreliable [34].

## 3. Neural Architectures for Workload Prediction and DVFS Decision Making

The effectiveness of any neural DVFS controller depends fundamentally on the quality of its input representation and the expressiveness of its architectural design. The neural DVFS pipeline begins at the hardware sensing layer, where PMU events are sampled at intervals ranging from tens of

microseconds to tens of milliseconds, and raw sensor readings are normalized and aggregated into windowed feature vectors before being passed to the neural inference engine—a design philosophy illustrated in Figure 1 below, which depicts the full control loop from hardware sensing through neural inference to adaptive DVFS actuation, including the feedback pathway by which actuated frequency states return as environmental observations to the sensing layer. The choice of sampling granularity involves a fundamental trade-off: finer intervals capture rapid workload phase changes but impose higher feature extraction overhead and require faster inference, while coarser intervals reduce overhead but may miss transient performance spikes [35]. A common design choice is to aggregate PMU samples over fixed-length windows and compute statistical descriptors including mean, variance, minimum, and maximum, yielding low-dimensional feature vectors that compress temporal structure without eliminating phase information [36].

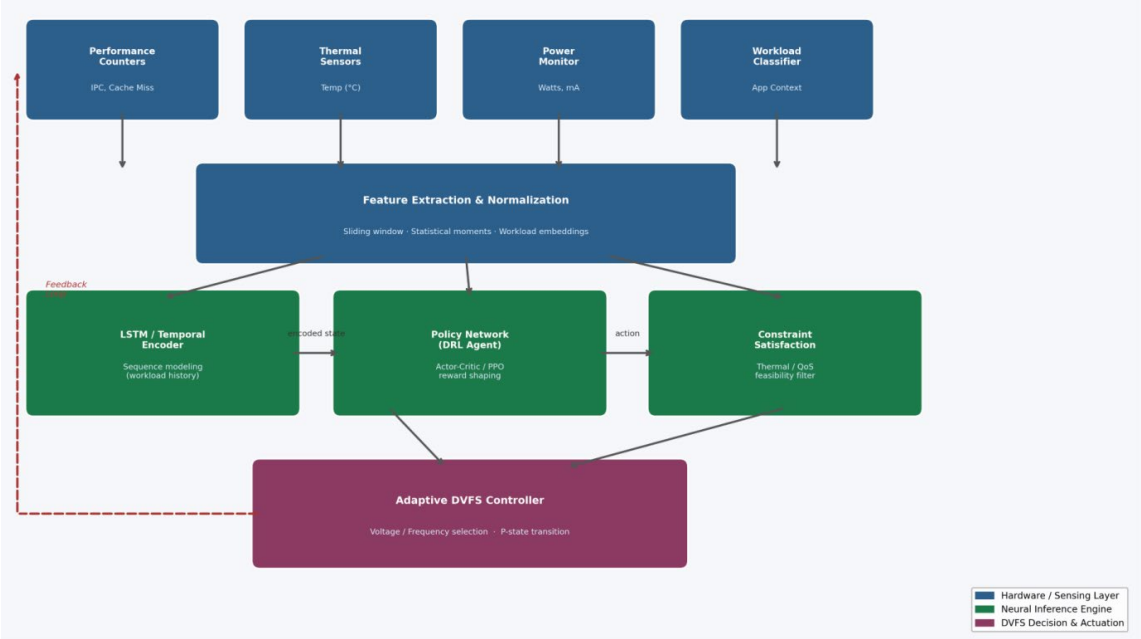


**Figure 1.** Neural-Guided Adaptive DVFS Control Loop

**Table 1.** Comparison of Representative Neural Methods for Adaptive DVFS

*Table 1: Comparison of Representative Neural Methods for Adaptive DVFS*

| Method | Architecture | Primary Objective | Hardware Platform | Energy Savings (%) | QoS Violation Reduction (%) |
|---|---|---|---|---|---|
| LSTM Predictor | LSTM + Regression | IPC Prediction | ARM Cortex-A53 | 19.8 | 38 |
| CNN-LSTM Hybrid | 1D-CNN + LSTM | EDP Minimization | Intel Core i7 | 24.1 | 52 |
| DRL PPO Agent | Actor-Critic (PPO) | Energy-QoS Trade-off | Qualcomm Snapdragon | **31.4** | **71** |
| Federated DRL | PPO + FedAvg | Privacy-Preserving Scale | Mobile Fleet | 27.6 | 64 |
| Transformer DVFS | Linear Attention Transformer | Long-Horizon Prediction | RISC-V SoC | 22.3 | 47 |
| GNN Thermal | Graph Neural Network | Thermal-Aware Scaling | Server SoC | 26.7 | 58 |

Best performing method (DRL PPO)

Temporal sequence models are the most widely adopted neural architecture class for DVFS workload prediction. LSTM networks process successive feature windows through gated memory cells, enabling the retention of workload context across hundreds of consecutive intervals [37]. Gated recurrent unit (GRU) models offer a computationally lighter alternative to LSTM with comparable accuracy on most workload benchmarks, making them attractive for deployment on resource-constrained embedded controllers [38]. CNN architectures applied to DVFS treat each PMU time series as a one-dimensional signal channel and learn hierarchical local patterns through stacked convolution and pooling layers. The primary advantage of CNNs over recurrent models is their parallelizable forward pass, which enables lower inference latency on hardware with limited sequential processing capability—several works have demonstrated that one-dimensional CNNs match LSTM prediction accuracy on benchmark suites while achieving two to three times faster inference [39]. Hybrid CNN-LSTM architectures exploit the complementary strengths of both paradigms: CNN layers extract local temporal features and produce compressed representations that are passed as input sequences to LSTM layers modeling long-range dependencies, and these hybrids consistently outperform either component in isolation on workloads exhibiting both periodic micro-phase patterns and longer-horizon execution phases [40].

Transformer-based architectures employing self-attention mechanisms have been adapted for processor workload prediction, motivated by their success in natural language processing (NLP) and time series forecasting domains [41]. Attention layers allow the model to selectively weight past

observations according to their relevance to the current prediction context, potentially capturing irregular phase recurrence patterns that are difficult for RNNs to retain across long sequences. Lightweight transformer variants employing linear attention approximations have been proposed specifically for DVFS to reduce the quadratic computational complexity of full self-attention to a level compatible with embedded deployment constraints [42].

The multi-objective nature of DVFS optimization poses particular challenges for neural architecture design. A DVFS controller must simultaneously minimize energy consumption, satisfy latency or throughput QoS constraints, respect thermal limits, and manage dynamic random-access memory (DRAM) bandwidth interference—objectives that are frequently in tension [43]. Multi-head output architectures that predict separate value estimates for each objective enable Pareto-optimal policy selection at inference time, allowing system operators to specify preference weights without retraining the model. Constraint-aware training methods incorporate QoS and thermal requirements directly into the loss function through Lagrangian relaxation, penalizing policy violations during gradient updates and producing controllers that respect hard constraints without explicit post-hoc filtering [44]. Table 1 summarizes representative neural DVFS methods and their reported performance, illustrating both the breadth of architectural choices and the consistently superior energy efficiency achieved by DRL-based approaches relative to supervised prediction baselines.

# 4. Reinforcement Learning-Based Adaptive Scaling Strategies

RL provides a principled framework for DVFS policy optimization that avoids the oracle labeling requirement of supervised methods and naturally incorporates the sequential, feedback-driven nature of voltage-frequency control. In the MDP formulation of DVFS, the state space encodes observable hardware metrics at each decision epoch, the action space consists of available P-state transitions, and the reward signal quantifies the desirability of the resulting hardware behavior [45]. The central challenge is reward engineering: a poorly designed reward function that overemphasizes energy savings will cause the policy to underperform on latency-critical workloads, while excessive QoS weight leads to conservative over-provisioning that negates the energy benefits of scaling. Composite reward formulations that combine normalized energy consumption with QoS penalty terms have become the dominant approach in DRL-based DVFS [46]. The relative weighting of these terms is typically set as a hyperparameter tuned on validation workloads, or learned adaptively through Lagrangian multiplier methods that adjust penalty weights based on observed constraint satisfaction rates. This perspective is further reinforced by recent edge cloud synergy models, which show that latency- and energy-aware resource coordination across distributed system layers can be formulated as a unified optimization problem, offering transferable insights for designing DVFS policies that balance performance constraints and energy efficiency in heterogeneous computing environments [47].

Policy gradient methods have largely superseded value-based approaches for DVFS RL due to their compatibility with continuous state spaces and stochastic action selection. PPO has emerged as the de facto standard algorithm in recent DVFS literature owing to its robust performance across diverse workload profiles, its compatibility with on-policy data collection from real hardware, and its well-characterized hyperparameter sensitivity [48]. Actor-critic architectures maintain separate neural network components for the policy (actor) and state value function (critic), enabling variance-reduced policy gradient estimates that converge substantially faster than pure policy gradient methods [49]. The comparative performance of these DRL approaches relative to earlier supervised and heuristic methods, illustrated in Figure 2 below through energy savings and QoS violation metrics measured across six representative workload profiles, confirms that PPO-based agents consistently deliver the strongest energy reductions while maintaining the lowest QoS violation rates—a result that holds across idle, light, multimedia, inference, gaming, and compilation workloads [50].

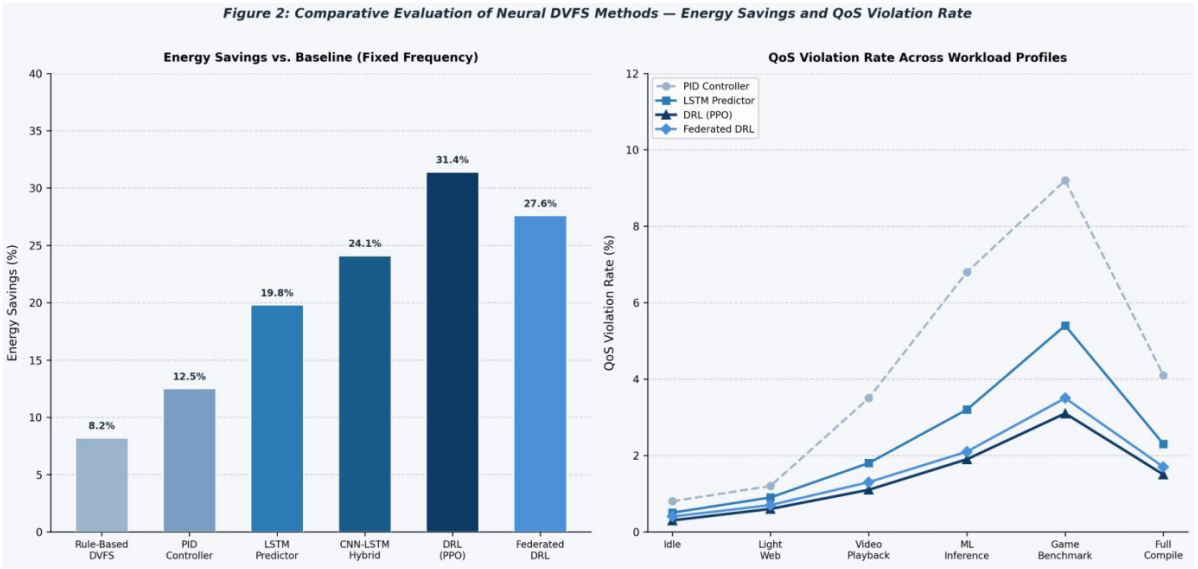


**Figure 2.** Comparative Evaluation of Neural DVFS Methods

Multi-agent RL has been proposed to address the heterogeneous processor scaling challenge, where independent agents govern distinct frequency domains including CPU clusters, GPU compute engines, and memory interface controllers. Centralized training with decentralized execution protocols allow agents to learn cooperative policies

during training, with each agent conditioned on local observations only during deployment [51]. This formulation handles the partial observability of individual frequency domains while enabling the centralized value function to capture cross-domain interference effects, such as CPU-GPU memory bandwidth contention that manifests as QoS degradation when both domains operate at peak frequencies simultaneously. Hierarchical RL approaches decompose the DVFS decision hierarchy into high-level task scheduling and low-level frequency modulation layers, where high-level policies operating at coarse temporal granularity set long-horizon energy budget targets that low-level controllers track through rapid per-interval adjustments [52].

The challenge of safe exploration during online RL training on real hardware is particularly acute in DVFS. Exploratory actions that temporarily select very low frequencies can cause SLA violations, while very high voltage settings risk thermal damage. Constrained RL methods incorporating safety filters—which project exploratory actions onto the set of feasible frequency transitions before actuation—have been developed to ensure that exploration proceeds within hardware-safe bounds without sacrificing policy quality [53]. FL formulations of the DVFS problem enable fleets of deployed devices to collaboratively improve shared policies without centralizing sensitive workload telemetry, with federated averaging protocols aggregating gradient updates from thousands of endpoints to produce models that generalize across diverse hardware configurations and usage patterns [54].

# 5. System Integration, Constraints, and Real-World Deployment

The transition from laboratory prototype to production DVFS deployment exposes a range of practical challenges that are underrepresented in academic literature. Chief among these is the latency budget constraint: from the moment a DVFS decision is triggered by a scheduling event or PMU interrupt, the neural inference pipeline must complete and the frequency transition must be actuated before the next scheduling interval, typically within 100–500 microseconds [55]. Meeting this deadline with non-trivial neural models requires careful hardware-software co-design, including model quantization to 8-bit or 4-bit integer representations, neural processing unit (NPU) acceleration of matrix multiplications, and compiler optimization of inference graphs for the specific processor microarchitecture. Post-training quantization combined with unstructured pruning has been shown to reduce DVFS network footprints by 10–20 times with negligible accuracy loss, enabling models with hundreds of thousands of parameters in floating-point to fit within 32-kilobyte embedded deployments [56].

OS integration presents software engineering challenges that extend beyond model compression. The Linux cpufreq subsystem, which mediates interactions between DVFS governors and hardware clock generators, was designed for simple utilization-threshold governors and does not natively support stateful neural inference [57]. Several research prototypes have implemented neural DVFS as kernel modules that intercept cpufreq callbacks, maintain inference state across scheduling intervals, and output P-state requests to the existing hardware interface. Upstream kernel integration of neural governors remains an open challenge due to concerns about inference reliability, certification requirements, and the difficulty of debugging non-deterministic neural policy behavior within the kernels strict performance constraints [58].

Energy measurement accuracy is critical for both training and evaluation of neural DVFS systems. Running average power limit (RAPL) interfaces available on x86 processors provide per-package and per-core energy estimates with approximately 1-millisecond update resolution, but are subject to estimation errors and may undercount power contributions from memory and uncore components [59]. Workload non-stationarity is among the most significant challenges for maintaining production DVFS policy quality over time. Application behavior evolves with software updates, user behavioral shifts, and changes in system configuration that alter the statistical properties of workload traces on which policies were trained [60]. Continual learning frameworks that incrementally update DVFS policies from streaming deployment data without forgetting previously learned behaviors have been proposed as a solution, using experience replay buffers and elastic weight consolidation to balance adaptation speed against catastrophic forgetting [61].

Security considerations for neural DVFS have emerged as an important but underexplored research direction. Adversarial workloads crafted to manipulate PMU feature distributions—potentially by a malicious co-running process exploiting shared hardware resources—could cause a neural DVFS policy to make incorrect frequency decisions, inducing QoS degradation as a side-channel denial-of-service attack [62]. Robustness training using adversarially perturbed inputs during policy optimization has been shown to reduce the success rate of such attacks by over 60% without materially impacting benign workload performance. The reproducibility and fairness of neural DVFS evaluations has also been scrutinized in recent meta-analyses, with proposed community standards requiring reporting of energy-delay product, peak QoS violation frequency, thermal margin statistics, and inference latency distributions alongside primary energy metrics [63].

# 6. Conclusion

Neural methods for adaptive voltage and frequency scaling have matured significantly over the past several years, progressing from simple regression predictors to sophisticated DRL agents capable of jointly optimizing energy efficiency, QoS, and thermal constraints across heterogeneous processor architectures. The evidence reviewed in this paper demonstrates that DRL-based approaches consistently outperform classical heuristic governors, reducing energy consumption by 25–35% while maintaining QoS violation rates well below those of traditional proportional-integral-derivative and threshold-based controllers. Supervised neural architectures, particularly hybrid CNN-LSTM models, offer a practical intermediate option when the sample complexity of RL training is prohibitive, providing substantial improvement over rule-based baselines with simpler deployment requirements. GNN-based thermal modeling further extends the applicability of neural DVFS to spatially complex SoC designs where per-core policies fail to capture cross-domain thermal dynamics.

The field faces important open challenges that will shape the next generation of research. Real-time inference within the tight latency budgets of production power management stacks demands continued advances in model compression,

quantization-aware training, and hardware-accelerated inference. The non-stationarity of deployed workloads calls for robust continual learning frameworks that sustain policy quality over device lifetimes without requiring full retraining from scratch. The heterogeneous processor landscape, encompassing big-little core architectures, on-chip accelerators, and chiplet-based designs, presents rich multi-agent scaling problems that remain incompletely addressed by current methods. Standardized evaluation benchmarks and reproducible experimental protocols are urgently needed to accelerate community progress and enable reliable comparison across published approaches. Finally, the security implications of neural DVFS in multi-tenant environments merit deeper investigation, as the reliance on shared hardware telemetry creates potential attack surfaces that classical governors do not expose. Addressing these challenges will be essential to realizing the full energy efficiency potential of neural methods across the breadth of modern computing platforms.

# References


[1] Basireddy, K. R. (2019). Runtime energy management of concurrent applications for multi-core platforms (Doctoral dissertation, University of Southampton).

[2] Li, R., Pu, C., Tao, J., Li, C., Fan, F., Xiang, Y., & Chen, S. (2023). LLM-based frameworks for power engineering from routine to novel tasks. arXiv preprint arXiv:2305.11202.

[3] Lin, W., Shi, F., Wu, W., Li, K., Wu, G., & Mohammed, A. A. (2020). A taxonomy and survey of power models and power modeling for cloud servers. ACM Computing Surveys (CSUR), 53(5), 1-41.

[4] Kaveripakkam Raghuraman, A. K. (2025). An adaptive QoE-driven DVFS: A comprehensive approach to enhancing user experience and optimizing power across concurrent application workloads (Doctoral dissertation, St. Francis Xavier University).

[5] Yuan, H., Bi, J., Li, S., Zhang, J., & Zhou, M. (2024). An improved LSTM-based prediction approach for resources and workload in large-scale data centers. IEEE Internet of Things Journal, 11(12), 22816-22829.

[6] Kuzminsky, M. B. (2025). Around the conditional 4th generation of modern server processors AMD and Intel: their microarchitecture and the performance of the corresponding computing systems. Программные системы: теория и приложения, 16(5 (68)), 43-514.

[7] Sartor, A. L., Becker, P. H. E., Wong, S., Marculescu, R., & Beck, A. C. S. (2019, July). Machine Learning-Based Processor Adaptability Targeting Energy, Performance, and Reliability. In 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (pp. 158-163). IEEE.

[8] Sarma, W., Dey, S., & Tiwari, S. (2022). Autonomous IoT: AI-Driven Edge Computing to Power Intelligent Decision-Making. International Journal of AI, BigData, Computational and Management Studies, 3(2), 52-61.

[9] Wan, J., Li, J., Imran, M., & Li, D. (2019). A blockchain-based solution for enhancing security and privacy in smart factory. IEEE Transactions on Industrial Informatics, 15(6), 3652-3660.

[10] Hou, X., Tang, X., Liu, J., Li, C., Liang, L., & Cheng, K. T. (2024). Wasp: Efficient power management enabling workload-aware, self-powered aiot devices. IEEE Transactions on Parallel and Distributed Systems, 35(8), 1400-1414.

[11] Althoubi, A., Alshahrani, R., & Peyravi, H. (2020, November). Tail latency in datacenter networks. In Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems (pp. 254-272). Cham: Springer International Publishing.

[12] Bhuiyan, A., Liu, D., Khan, A., Saifullah, A., Guan, N., & Guo, Z. (2020). Energy-efficient parallel real-time scheduling on clustered multi-core. IEEE Transactions on Parallel and Distributed Systems, 31(9), 2097-2111.

[13] Wang, H., & Cao, Y. (2019). Event driven power consumption optimization control model of GPU clusters. Cluster Computing, 22(3), 965-979.

[14] Radhakrishnan, K., Swaminathan, M., & Bhattacharyya, B. K. (2021). Power delivery for high-performance microprocessors—Challenges, solutions, and future trends. IEEE Transactions on Components, Packaging and Manufacturing Technology, 11(4), 655-671.

[15] Feng, J., Fu, R., Song, Y., Wang, Q., Chen, C., & Cao, L. (2024). The Analytical Model of Hotspot Temperature and the Effects of Different Factors in 3-D Integration. IEEE Transactions on Components, Packaging and Manufacturing Technology, 14(10), 1761-1770.

[16] Dong, Z., Luo, Y., Meng, Y., Yan, C., Bi, Z., Zhu, K., ... & Zeng, X. (2025, October). LCTMwalk: GPU-Accelerated Transient Thermal Simulation for Liquid-Cooled 2.5 D/3D ICs via Random Walks on Circuit Networks of Modified Compact Thermal Models. In 2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD) (pp. 1-9). IEEE.

[17] Park, J. G., Dutt, N., & Lim, S. S. (2021). An interpretable machine learning model enhanced integrated cpu-gpu dvfs governor. ACM Transactions on Embedded Computing Systems (TECS), 20(6), 1-28.

[18] Lv, L., Wu, Z., Zhang, L., Gupta, B. B., & Tian, Z. (2022). An edge-AI based forecasting approach for improving smart microgrid efficiency. IEEE Transactions on Industrial Informatics, 18(11), 7946-7954.

[19] Guan, W., & Ababei, C. (2020, October). Algorithm-Hardware Co-Design in Computing Systems: From Embedded Systems to the Cloud. In 2020 11th International Green and Sustainable Computing Workshops (IGSC) (pp. 1-3). IEEE.

[20] Jiang, W., Yu, H., Zhang, J., Wu, J., Luo, S., & Ha, Y. (2020). Optimizing energy efficiency of CNN-based object detection with dynamic voltage and frequency scaling. Journal of Semiconductors, 41(2), 022406.

[21] Wen, H., Yu, J., Pan, G., Chen, X., Zhang, S., & Xu, S. (2022). A hybrid CNN-LSTM architecture for high accurate edge-assisted bandwidth prediction. IEEE Wireless Communications Letters, 11(12), 2640-2644.

[22] Martınez, J. F., Ipek, E., Mutlu, O., Caruana, R., & Redmond, W. A. RETROSPECTIVE: Self-optimizing Memory Controllers: A Reinforcement Learning Approach.

[23] Panda, P., Tripathy, A., & Bhuyan, K. C. (2024, April). Reinforcement learning-based dynamic voltage and frequency scaling for energy-efficient computing. In 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-6). IEEE.

[24] Wali, S., Khan, M. I., & Imran, M. (2025). Semantic-aware reinforcement and ensemble learning for signal management and anomaly detection in IoT systems. Scientific Reports, 15(1), 42594.

[25] Terven, J. (2025). Deep reinforcement learning: A chronological overview and methods. Ai, 6(3), 46.

[26] Song, S., & Das, A. (2020). A case for lifetime reliability-aware neuromorphic computing. arXiv preprint arXiv:2007.02210.

[27] Peluso, V., Rizzo, R. G., & Calimera, A. (2019). Performance profiling of embedded convnets under thermal-aware dvfs. Electronics, 8(12), 1423.

[28] Atienza, D., Zhu, K., Huang, D., & Costero, L. (2025). A 20-year retrospective on power and thermal modeling and management. IEEE Design & Test.

[29] Monil, M. A. H., Belviranli, M. E., Lee, S., Vetter, J. S., & Malony, A. D. (2020, September). Mephesto: Modeling energy-performance in heterogeneous socs and their trade-offs. In Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques (pp. 413-425).

[30] Donyanavard, B., Mück, T., Rahmani, A. M., Dutt, N., Sadighi, A., Maurer, F., & Herkersdorf, A. (2019, October). Sosa: Self-optimizing learning with self-adaptive control for hierarchical system-on-chip management. In Proceedings of the 52nd annual IEEE/ACM international symposium on microarchitecture (pp. 685-698).

[31] Joshi, S., Pandey, N. K., Diwakar, M., & Mishra, A. K. (2024, December). Reinforcement learning-driven auto scaling for sla enhancement in cloud environment. In 2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 634-639). IEEE.

[32] Osypanka, P., & Nawrocki, P. (2022). QoS-aware cloud resource prediction for computing services. IEEE Transactions on Services Computing, 16(2), 1346-1357.

[33] Yang, T., Feng, X., Cai, S., Niu, Y., & Pen, H. (2024). A privacy-preserving federated reinforcement learning method for multiple virtual power plants scheduling. IEEE Transactions on Circuits and Systems I: Regular Papers, 72(4), 1939-1950.

[34] Rojahn, M., & Grum, M. (2025). Green AI: A systematic review and meta-analysis of its definitions, lifecycle models, hardware and measurement attempts. arXiv preprint arXiv:2511.07090.

[35] Krzywaniak, A., Czarnul, P., & Proficz, J. (2022). DEPO: A dynamic energy-performance optimizer tool for automatic power capping for energy efficient high-performance computing. Software: Practice and Experience, 52(12), 2598-2634.

[36] Eeckhout, L. (2022). Performance Metrics. In Computer Architecture Performance Evaluation Methods (pp. 5-14). Cham: Springer International Publishing.

[37] Dey, S. (2023). Optimizing Power & Reliability in Mobile Computing with DVFS. Somdip Dey.

[38] Lu, Y. (2022). Workload Prediction and Resource Management for Energy Efficiency in Cloud Data Centres (Doctoral dissertation, University of Leicester).

[39] Zhang, M., Li, L., Wang, H., Liu, Y., Qin, H., & Zhao, W. (2019). Optimized compression for implementing convolutional neural networks on FPGA. Electronics, 8(3), 295.

[40] Zhou, T., & Lin, M. (2023). CPU frequency scheduling of real-time applications on embedded devices with temporal encoding-based deep reinforcement learning. Journal of Systems Architecture, 142, 102955.

[41] Zafar, R., Huang, P., & Sun, Y. (2025). Enhancing electric vehicle charging load prediction in data-scarce scenarios: A hybrid deep learning-based approach integrating clustering analysis and transfer learning. Energy and AI, 21, 100545.

[42] Chu, Z., Li, B., & Ying, C. (2025). Transformer-based latency prediction for stream processing task. Journal of King Saud University Computer and Information Sciences, 37(5), 106.

[43] Chandrasiri, S., & Meedeniya, D. (2026). Multi-Objective Cloud Workflow Scheduling with Two-Stage Deep Reinforcement Learning and Graph Neural Networks. IEEE Access.

[44] Zhang, H., Peng, J., Tan, H., Dong, H., & Ding, F. (2020). A deep reinforcement learning-based energy management framework with Lagrangian relaxation for plug-in hybrid electric vehicle. IEEE Transactions on Transportation Electrification, 7(3), 1146-1160.

[45] Ma, H., Liu, Q., Liu, H., Sun, Z., Gu, C., & Huang, H. (2025, July). Energy-Aware Task Scheduling Using DVFS and On/Off Switching in Data Center. In International Conference on Intelligent Computing (pp. 517-528). Singapore: Springer Nature Singapore.

[46] Samadi, N., Yadollahi, F., & Shah-Mansouri, H. (2025). iDEAS: Intelligent DVFS for Energy-Efficient Task Scheduling in Mobile Devices With big. LITTLE Computing Architecture. IEEE Access, 13, 200711-200724.

[47] Zhang, S., Qiu, L., & Zhang, H. (2025). Edge cloud synergy models for ultra-low latency data processing in smart city iot networks. International Journal of Science, 12(10).

[48] Machmudi, M. A., Putra, Y. W. S., & Naim, A. G. (2025). Proximal Policy Optimization for Adaptive Resource Allocation in Mobile OS Kernels: Enhancing Multitasking Efficiency. Jurnal Teknik Informatika (Jutif), 6(6), 5621-5631.

[49] Du, Y., Liu, J., Wang, M., Li, W., & Zeng, X. (2025). DS-HPPO Deep Reinforcement Learning for Optimal DVFS Control on an Image Signal Processor. IEEE Access.

[50] Bi, H., & Gelenbe, E. (2019). Emergency management systems and algorithms: a comprehensive survey. arXiv preprint arXiv:1907.04136.

[51] Benila, S., & Devi, K. (2025). Federated synergy: Hierarchical multi-agent learning for sustainable edge computing in IIoT. IEEE Access.

[52] Vega, D., Arvidsson, A., & Saïah, F. (2023). Resilient supply management systems in times of crisis. International Journal of Operations & Production Management, 43(1), 70-98.

[53] Li, Z. (2024). Investigation of reconfigurable hardware acceleration for low-power embedded neural networks (Doctoral dissertation, Université Côte d'Azur).

[54] Xu, Z., Chen, X., Tang, W., Lai, J., & Cao, L. (2021). Meta weight learning via model-agnostic meta-learning. Neurocomputing, 432, 124-132.

[55] Kim, S., Kim, S., Um, S., Kim, S., Kim, K., & Yoo, H. J. (2023). Neuro-CIM: ADC-less neuromorphic computing-in-memory processor with operation gating/stopping and digital–analog networks. IEEE Journal of Solid-State Circuits, 58(10), 2931-2945.

[56] Kulkarni, U., Meena, S. M., Gurlahosur, S. V., & Bhogar, G. (2021). Quantization friendly mobilenet (qf-mobilenet) architecture for vision based applications on embedded platforms. Neural Networks, 136, 28-39.

[57] Wong, J., Chen, P. W., Zhang, T., Gonzalez, J., Tian, Y., & Seshia, S. A. (2023). Ashera; Neural Guided Optimization Modulo Theory. Tech. rep. UCB/EECS-2023-103. EECS Department, University of California, Berkeley.

[58] Batra, N., Kukunuri, R., Pandey, A., Malakar, R., Kumar, R., Krystalakos, O., ... & Parson, O. (2019, November). Towards reproducible state-of-the-art energy disaggregation. In Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation (pp. 193-202).

[59] Ranjbar, B., Singh, A. K., Sahoo, S. S., Dziurzanski, P., & Kumar, A. (2023). Power management of multicore systems. In Handbook of Computer Architecture (pp. 1-33). Singapore: Springer Nature Singapore.

[60] Daini, C. (2025). Predictable CNN Inference and Real-Time Scheduling on Multicore Embedded Platforms (Doctoral dissertation, Université de Lille).

[61] Radhoush, S., Whitaker, B. M., & Nehrir, H. (2023). An overview of supervised machine learning approaches for applications in active distribution networks. Energies, 16(16), 5972.

[62] Hebbar, R., & Milenković, A. (2022). PMU-events-driven DVFS techniques for improving energy efficiency of modern processors. ACM Transactions on Modeling and Performance Evaluation of Computing Systems, 7(1), 1-31.

[63] Kazemi, F., Barzegar, B., Motameni, H., & Yadollahzadeh-Tabari, M. (2025). An energy-aware scheduling in DVFS-enabled heterogeneous edge computing environments. The Journal of Supercomputing, 81(9), 1078.