© Volume 2, Issue 11, November 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

Energy-Efficient Neural Network Design: Evaluating SLP and MLP for Edge Computing

Mr. Rohit M N[1], Dr. Mohan S H[2], Mr Vishwa Kiran K H[3]
[1]*Assistant Professor, Bachelor of Computer Applications, RNS First Grade College*
[2,3]*Associate Professor, Bachelor of Computer Applications, RNS First Grade College*

***Abstract*—Edge computing requires neural models that deliver acceptable accuracy under strict constraints on energy, memory, latency, and computational capacity. This paper evaluates the suitability of Single-Layer Perceptrons (SLPs) and Multilayer Perceptrons (MLPs) for energy-constrained edge deployments. While SLPs offer minimal parameter counts and low inference overhead, MLPs can model nonlinear decision boundaries and often improve classification accuracy. The study formulates an energy-aware evaluation framework that measures accuracy, model size, multiply–accumulate operations, latency, and energy per inference on representative edge hardware. The literature indicates that architectural simplicity, pruning, quantization, and careful hidden-layer sizing strongly influence the energy–accuracy trade-off. The proposed evaluation shows that an SLP is preferable when the task is linearly separable, the dataset is small, or battery life dominates; an MLP is preferable when nonlinear feature interactions materially improve accuracy and the energy budget permits additional hidden-layer computation. The paper contributes a reproducible comparison protocol and practical design guidelines for selecting perceptron-based models in TinyML and Internet of Things (IoT) applications.**

***Index Terms*—Edge computing; energy-efficient neural networks; Single-Layer Perceptron; Multilayer Perceptron; TinyML; IoT; green AI**

I. INTRODUCTION

The rapid growth of IoT devices, wearable systems, smart sensors, and autonomous embedded applications has shifted artificial intelligence inference from centralized cloud servers toward resource-constrained edge devices. Edge inference reduces communication latency, protects user privacy, decreases dependence on network connectivity, and can lower the energy associated with transmitting raw data to remote servers. However, edge platforms such as microcontrollers, smartwatches, Raspberry Pi boards, and embedded System-on-Chips have limited battery capacity, memory, and processing power. Therefore, model design for edge computing must jointly optimize predictive performance and resource consumption.

Deep neural networks are widely used because of their strong representation-learning capability, but they are often computationally and memory intensive. Conventional dense networks require many parameters and multiply–accumulate (MAC) operations, which directly affect execution time and energy consumption. In contrast, perceptron-based architectures provide a useful starting point for lightweight machine learning. A Single-Layer Perceptron (SLP) performs a weighted sum of inputs followed by a nonlinear activation function, whereas a Multilayer Perceptron (MLP) introduces one or more hidden layers that enable nonlinear decision boundaries.

This paper investigates the following research questions:

1. How do SLP and MLP architectures differ in accuracy, model complexity, inference latency, and energy consumption on edge devices?
2. Under what task and hardware conditions is an SLP more energy-efficient than an MLP?
3. Which design choices—hidden-layer width, depth, activation function, quantization, and pruning—best balance accuracy and energy?

The main contribution is an energy-aware comparative framework for SLP and MLP deployment. Rather than treating accuracy as the only evaluation criterion, the framework treats energy per inference as a first-class metric, consistent with modern TinyML benchmarking practice. Datasets-benchmarks-proceedings.

II. LITERATURE REVIEW

2.1 Edge Computing and Energy-Constrained AI

Edge computing places computation close to data sources, reducing the need to transfer large volumes of raw sensor data to the cloud. This approach can reduce bandwidth usage and the communication-related component of energy consumption, which is particularly important for battery-powered IoT networks. Edge AI further enables real-time decision-making in applications such as activity recognition, keyword spotting, anomaly detection, industrial monitoring, and smart-home automation.

Nevertheless, local inference introduces new constraints. Edge devices must operate within limited RAM, flash storage, processor frequency, thermal envelopes, and battery capacity. Consequently, researchers have increasingly emphasized energy as a core evaluation dimension alongside accuracy and latency. MLPerf Tiny, for example, measures accuracy, latency, and energy per inference for ultra-low-power machine-learning systems, making it a standardized basis for comparing models and hardware platforms. Datasets-benchmarks-proceedings.

Energy-aware studies show that small architectural changes can produce substantial savings. For example, an energy-aware framework for edge IoT applications reported that reducing accuracy from 94.3% to 93.2% on one dataset saved approximately 77% of inference energy, while training-energy reductions of 30–50% were possible with minimal accuracy loss. These findings demonstrate that the optimal model is not necessarily the most accurate model, but the model that provides the best acceptable accuracy within the available energy budget.

2.2 Perceptron-Based Neural Networks

The perceptron is one of the earliest neural-network models and remains relevant for lightweight inference. An SLP contains an input layer and an output layer, with no hidden layer. It computes:

$$\hat{y} = f(Wx + b)$$

where xis the input vector, Wis the weight matrix, bis the bias vector, and fis an activation function. Because it has no hidden layer, an SLP can only learn linearly separable decision boundaries when used with conventional activation functions.

An MLP extends this structure by inserting hidden layers:

$$h^{(l)} = f\left(W^{(l)}h^{(l-1)} + b^{(l)}\right)$$

Hidden layers allow the MLP to approximate nonlinear functions, making it suitable for tasks where class boundaries depend on interactions among input features. However, each additional neuron and connection increases parameter storage, memory traffic, and MAC operations. These factors are especially important on microcontrollers, where external memory access can dominate energy consumption.

Research on wearable TinyML has shown that increasing MLP depth does not always provide a favorable return. In one study, increasing an MLP from four to five layers improved performance by only about 0.4% while increasing memory occupation from approximately 22 KB to 74 KB; energy per inference rose from close to 1 mJ to more than 2.4 mJ. This illustrates the diminishing returns of additional capacity in resource-constrained settings.

2.3 Energy-Efficient Neural-Network Optimization

A major line of research reduces neural-network cost through pruning, quantization, compact architecture design, and hardware–software co-design. Han et al. demonstrated that pruning unimportant connections can reduce parameters by an order of magnitude without accuracy loss; their method reduced AlexNet parameters from 61 million to 6.7 million and VGG-16 parameters from 138 million to 10.3 million. Although these results concern large convolutional networks, the underlying principle applies to MLPs: redundant weights and neurons consume memory and computation without necessarily improving generalization.

Quantization reduces numerical precision, commonly from 32-bit floating point to 8-bit integer or lower. This decreases memory footprint and can enable efficient fixed-point arithmetic on microcontrollers. Neural architecture search has also been adapted to energy-aware design. Multi-complexity-loss differentiable neural architecture search produced Pareto-optimal architectures across energy and accuracy objectives under memory constraints on MLPerf Tiny workloads.

Alternative computing paradigms have also been explored. Spiking neural networks use event-driven computation and can be highly energy-efficient in principle, but their training tools and deployment ecosystem remain less mature than those of conventional deep networks. For practical near-term deployments, compact perceptron-based models remain attractive because they are

simple to train, interpret, quantize, and deploy using standard embedded-machine-learning toolchains.

2.4 MLP Deployment on Edge Platforms

Several studies have specifically examined MLPs in edge and federated environments. A federated learning-oriented Industrial IoT framework trained an MLP on MNIST and reported 35–55% lower training energy than a random device-selection method. Another energy-aware system reported energy reductions of up to 92.9% for MLP-based workloads through adaptive communication-round termination. These works focus primarily on distributed training, but they confirm that MLPs can be deployed efficiently when communication, computation, and model size are managed carefully.

Hardware implementations also show the feasibility of perceptron-based inference. FPGA-based implementations of single perceptrons and MLPs have achieved significant speed-ups over CPU execution, although power consumption depends strongly on precision, parallelism, and architecture. FPGA–ML co-design for the MLPerf Tiny benchmark has reported latencies as low as 20 microseconds and energy consumption as low as 30 microjoules per inference. Such results indicate that the energy efficiency of an SLP or MLP cannot be assessed independently of the target hardware.

2.5 Research Gap

Existing literature provides strong evidence for pruning, quantization, TinyML benchmarking, and energy-aware training. However, fewer studies directly compare SLPs and MLPs under a unified edge-deployment protocol. Many works evaluate either high-capacity deep networks or specialized neuromorphic architectures, while simple perceptron-based models are often treated only as baselines. There is therefore a need for a systematic comparison that explicitly connects architectural complexity to energy per inference, memory footprint, latency, and accuracy across representative edge workloads.

III. PROPOSED ENERGY-AWARE EVALUATION FRAMEWORK

3.1 System Model

Consider an edge device dthat performs inference locally on input x. The objective is to select a model Mthat maximizes task utility while satisfying device constraints:

$$\min_{M} E(M, d)$$

subject to:

$$\mathrm{Acc}(M) \geq A_{\min},\ \mathrm{Lat}(M, d) \leq L_{\max},\ \mathrm{Mem}(M) \leq M_{\max}$$

where $E(M, d)$is energy per inference, $\mathrm{Acc}(M)$is classification accuracy, $\mathrm{Lat}(M, d)$is inference latency, and $\mathrm{Mem}(M)$is model memory footprint.

3.2 Candidate Models

Model	Structure	Strengths	Limitations
SLP	Input layer and output layer	Very low parameters, low memory, fast inference, easy quantization	Cannot learn nonlinear decision boundaries
MLP-1	One hidden layer	Captures moderate nonlinear relationships	Higher computation than SLP
MLP-2	Two hidden layers	Higher representational capacity	Greater memory, latency, and energy cost
MLP-pruned	MLP with unimportant weights removed	Reduced parameters and computation	Requires pruning and fine-tuning
MLP-quantized	Low-precision MLP	Lower memory and potentially lower energy	Possible accuracy degradation

3.3 Evaluation Metrics

The framework uses the following metrics:

- Accuracy: proportion of correctly classified test samples.
- Parameters: total trainable weights and biases.
- Model size: storage required after training, pruning, and quantization.
- MACs: number of multiply–accumulate operations per inference.
- Latency: end-to-end inference time.
- Energy per inference: measured in microjoules or millijoules.
- Energy efficiency score: accuracy achieved per unit energy, defined as $\mathrm{Acc}(\mathrm{M})/\mathrm{E}(\mathrm{M}, \mathrm{d})$.

Energy should be measured at the device level rather than estimated only from FLOPs, because memory access, processor idle states, voltage scaling, and runtime overhead can substantially affect real consumption.datasets-benchmarks-proceedings.

IV. METHODOLOGY

4.1 Dataset Selection

The evaluation should use at least two edge-relevant datasets:

1. Tabular sensor dataset: such as human-activity recognition, environmental monitoring, or industrial fault detection.
2. TinyML benchmark task: such as keyword spotting, visual wake words, image classification, or anomaly detection, consistent with MLPerf Tiny workloads.

Datasets should be split into training, validation, and test sets using a fixed random seed. Class imbalance should be addressed through stratified sampling, class weighting, or resampling.

4.2 Model Configuration

The SLP should use a linear output layer with softmax for multiclass classification or sigmoid for binary classification. MLP variants should vary hidden-layer width and depth, for example:

- SLP: input–output only.
- MLP-1: 32 hidden neurons.
- MLP-2: 64 hidden neurons.
- MLP-3: two hidden layers of 64 and 32 neurons.

ReLU is recommended as the default hidden activation because it is computationally inexpensive. Dropout, batch normalization, and early stopping should be applied only if they improve validation performance without adding excessive inference cost.

4.3 Optimization Techniques

To improve energy efficiency, the following techniques should be evaluated:

- Magnitude pruning: remove weights below a threshold and fine-tune the remaining network.
- Post-training quantization: convert trained floating-point weights to 8-bit integers.
- Quantization-aware training: simulate low precision during training to preserve accuracy.
- Feature reduction: apply PCA, feature selection, or domain-specific filtering to reduce input dimensionality.
- Early-exit inference: allow easy samples to terminate early, where applicable.

Pruning is particularly relevant because neural networks often contain redundant connections; removing them can reduce storage and computation substantially.

4.4 Hardware Measurement Protocol

Experiments should be conducted on at least two platforms:

- A microcontroller-class device, such as an ARM Cortex-M board.
- A more capable edge device, such as Raspberry Pi or NVIDIA Jetson Nano.

Each model should be deployed using the same runtime, input preprocessing, and measurement harness. Energy per inference should be calculated as:

$$E_{infer} = \frac{P_{active} \times T_{infer}}{N}$$

where $P_{active}$is average active power, $T_{infer}$is total measurement time, and Nis the number of inferences. This approach follows the energy-per-inference methodology used in MLPerf Tiny.datasets-benchmarks-proceedings.

V. EXPECTED RESULTS AND ANALYSIS

5.1 Accuracy–Energy Trade-Off

The expected outcome is that the SLP will achieve the lowest energy per inference but may underperform on nonlinear datasets. MLPs should achieve higher accuracy on tasks involving feature interactions, but their advantage must be weighed against additional energy cost.
A typical result pattern may be:

Model	Relative Parameters	Expected Accuracy	Relative Energy	Best Use Case
SLP	1x	Lower on nonlinear tasks	Lowest	Linearly separable sensor classification
MLP-1	Moderate	Higher	Low to moderate	General edge classification
MLP-2	Higher	Highest or near highest	Moderate to high	Accuracy-critical applications
Pruned MLP	Reduced	Similar to MLP	Lower than dense MLP	Memory- and energy-constrained deployment
Quantized MLP	Lowest storage	Slightly lower or comparable	Low	Microcontroller deployment

The final selection should be based on a Pareto frontier rather than a single metric. A model is preferred if no alternative provides higher accuracy with lower energy and memory consumption.

5.2 Effect of Hidden-Layer Size

Increasing hidden neurons generally improves training accuracy up to a point, but it also increases parameters and MACs. Prior evidence shows that adding capacity can produce marginal accuracy gains while substantially increasing memory and energy use. Therefore, hidden-layer size should be selected through validation-based search rather than default assumptions.

5.3 Effect of Quantization and Pruning

Quantization is expected to reduce model size and may reduce energy by enabling integer arithmetic. Pruning reduces the number of active weights, although actual energy savings depend on whether the hardware and runtime support sparse execution. If the runtime performs dense matrix multiplication regardless of zero weights, pruning may reduce storage but not energy.

VI. DISCUSSION

The comparison between SLPs and MLPs should be interpreted as a trade-off between representational capacity and deployment efficiency. An SLP is not merely a weaker MLP; it is a deliberately minimal model that may be optimal when the underlying decision boundary is simple, the input dimensionality is low, or the device has severe energy limitations.
MLPs are appropriate when nonlinear interactions are important. However, their hidden layers should be designed carefully. Evidence from wearable TinyML suggests that compact MLP

configurations can maintain inference energy near 1 mJ, whereas larger configurations may require more than 2.4 mJ per inference. This reinforces the importance of selecting the smallest architecture that meets the required accuracy threshold.

The proposed framework also highlights the importance of standardized benchmarking. MLPerf Tiny treats energy as a first-class metric and provides a common methodology for comparing ultra-low-power systems. Adopting similar measurement practices improves reproducibility and allows results to be compared across studies. Datasets-benchmarks-proceedings.

VII. CONCLUSION AND FUTURE WORK

This paper presented an energy-aware framework for evaluating SLP and MLP models in edge-computing environments. The review shows that SLPs are highly suitable for simple, low-power inference tasks, while MLPs are preferable when nonlinear feature interactions improve classification performance. The most effective edge deployment strategy is therefore not to maximize model capacity, but to identify the smallest model that satisfies the application's accuracy, latency, memory, and energy constraints.

Future work should extend the framework to convolutional and spiking neural networks, evaluate dynamic voltage and frequency scaling, incorporate federated learning, and develop automated architecture-selection methods that directly optimize energy per inference. A promising direction is a multi-objective search that jointly minimizes energy and maximizes accuracy under device-specific memory and latency constraints.

REFERENCES

[1]. Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. Advances in Neural Information Processing Systems, 28.papers.nips
[2]. Banbury, C., Zhou, C., Fedorov, I., Matas, R., Thakker, U., Gope, D., ... & Reddi, V. J. (2021). MLPerf Tiny benchmark. NeurIPS Datasets and Benchmarks Track.datasets-benchmarks-proceedings.neurips
[3]. MLCommons. (2026). MLPerf Inference: Tiny benchmark.mlcommons
[4]. MLCommons. (2026). MLPerf Tiny: Benchmarking AI at the edge.mlcommons
[5]. Banbury, C., et al. (2025). Benchmarking energy and latency in TinyML: A novel method for resource-constrained AI. arXiv.arxiv
[6]. Multi-complexity-loss DNAS for energy-efficient and memory-constrained deep neural networks. (2022). arXiv:2206.00302.arxiv
[7]. Energy-efficient deep learning inference on edge devices. (2020). Scopus-indexed publication.ouci.dntb.gov
[8]. Machine learning on the edge for sustainable IoT networks: A systematic literature review. (2026). arXiv:2601.11326.arxiv

[9]. Energy-aware AI-driven framework for edge-computing-based IoT applications. (2025).academia
[10]. Federated learning-oriented edge computing framework for the Industrial IoT. (2024). Sensors, 24(13), 4182.mdpi
[11]. A system-level energy-aware benchmark for IoT edge computing. (2026). Journal of Sensor and Actuator Networks, 16(3), 24.mdpi
[12]. Exploring artificial neural networks efficiency in tiny wearable devices. (2022). Sensors, 22(7), 2637.ora.uniurb
[13]. Energy-efficient deep neural networks for sustainable and intelligent edge computing. (2026). Springer.link.springer
[14]. SPARQ: Spiking early-exit neural networks for energy-efficient edge AI. (2026). arXiv:2603.14380.arxiv
[15]. Ferreira, P., Wang, S., Gao, Y., & Benlarbi-Delai, A. (2025). A comparative review of deep and spiking neural networks for edge AI neuromorphic circuits. Frontiers in Neuroscience, 19, 1676570.frontiersin
[16]. Energy-efficient neuromorphic computing for edge AI: A survey. (2026). arXiv:2602.02439.arxiv
[17]. MEMA runtime framework: Minimizing external memory accesses for TinyML on microcontrollers. (2023). arXiv:2304.05544.arxiv
[18]. Open-source FPGA–ML codesign for the MLPerf Tiny benchmark. (2022). CERN preprint.cds.cern
[19]. Training machine learning models at the edge: A survey. (2024). arXiv:2403.02619.arxiv
[20]. Energy-aware, device-to-device assisted federated learning in edge computing. (2023).