INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

DYNAMIC MULTI-OBJECTIVE PRUNING OF DEEP NEURAL NETWORKS FOR ENERGY-EFFICIENT IOT EDGE COMPUTING

Asim Ud Din[1], Hizbullah Khalifa[2], Asad Ullah[*3], Sajjad Ahmad[4]

[1]*Department of Robotic and Artificial Intelligence, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST) University, Islamabad, Pakistan, 44000.*
[2]*Department of Computer Science, Government Postgraduate College (GPGC) Timergara, Khyber Pakhtunkhwa, Pakistan, 18300.*
[*3]*Institute of Computer Science and Information Technology, The University of Agriculture, Peshawar, Khyber Pakhtunkhwa, Pakistan, 25130.*
[4]*Department of Computer Science, School of Science and Technology, Nottingham Trent University, Nottingham, United Kingdom (UK), NG11 8NS.*

[*3]asadjn99@gmail.com

DOI: https://doi.org/10.5281/zenodo.23976765

Keywords
Evolutionary Computation, Internet of Things, Edge AI, Neural Network Pruning, NSGA-II, Energy Efficiency.

Article History
Received: 06 August 2026
Accepted: 20 October 2026
Published: 05 November 2026

Copyright @Author
Corresponding Author: *
Asad Ullah

Abstract
Deep Neural Networks (DNNs) are severely restricted from being deployed on Internet of Things (IoT) edge devices due to hardware constraints such as limited memory, low power and limited battery life. Model compression techniques reduce the storage overhead of deep learning models by pruning. However, conventional approaches do not take into account the changing energy levels within real-world IoT environments. To address this issue, this work presents a multi-objective evolutionary approach using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for dynamically pruned convolutional neural networks (CNNs). Structural filters and dense neurons are represented using binary encoding to allow for both maximal classification accuracy and minimal number of active parameters. Experiments were carried out using the UCI Human Activity Recognition (HAR) dataset, resulting in a clear Pareto Front of non-dominated solutions. The resulting pruned networks produced near 50% fewer active parameters while maintaining a high accuracy of 86.4%. Extreme levels of model compression achieved up to 66% of the baseline accuracy with very little energy consumption. This provides a powerful free solution for Edge AI to dynamically balance computation on resource-constrained IoT systems.

INTRODUCTION:

The fast development of the Internet of things (IoT) has pushed towards Edge AI which is a distributed computing framework that executes deep learning models at the edge level by running them directly on the embedded edge devices. This reduces inference time, protects user data privacy and prevents bandwidth congestion (Shi et al., 2016; Yu et al., 2018). However, modern Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) have millions of parameters that require extensive floating-point operations. These computationally heavy models placed onto edge devices that run on battery power cause the devices to rapidly run out of power and produce excess heat (Tmamna et al., 2024).

These can be addressed by developing 'Green Deep Learning' methods which reduce energy usage. Pruning methods have been developed to remove redundant weights and feature maps from the over-parametrized nature of neural networks, with little impact on model performance (Mishra & Gupta, 2023).

However, most pruning approaches are static. This means the model is pruned once before deployment, and cannot change during deployment. In dynamic scenarios such as energy harvested IoT, the amount of available power depends on many factors including: energy harvesting, temperature changes, and battery degradation. Thus, static models are not suited to this type of scenario.

In this paper we present a new method for multi-objective dynamic evolutionary pruning for IoT edge devices. Using the NSGA-II algorithm (Non-dominated Sorting Genetic Algorithm II) we automatically find Pareto optimal subnetworks that balance model accuracy with the number of active parameters (Cui et al., 2019). The pruned network family enables the edge device to adjust its level of complexity dynamically based on the remaining power.

LITERATURE REVIEW:

Deep learning inference has recently attracted much attention in academia when deploying within heterogeneous edge devices. Neural networks have traditionally increased in depth and complexity to improve accuracy. AlexNet (Krizhevsky et al., 2012), and ResNet (He et al., 2016) have both demonstrated that increasing depth can improve accuracy for more complex recognition tasks. Gradient based learning was implemented by LeCun et al. (1998). However, increasing depth creates an exponential increase in computational requirements (Szegedy et al., 2015). As these architectures do not meet the requirements of mobile/edge devices, there has been development within efficient network architectures. MobileNetV2 (Sandler et al., 2018) and MnasNet (Tan et al., 2019) were developed using inverted residual blocks and neural architecture search to reduce the number of operations.

However, due to strict energy requirements within edge devices, further optimization is required. Tmamna et al. (2024) categorized CNN pruning methods that can be used to create energy efficient models. They also note how new metrics must be developed to ensure multi-objectives are met for the model to have a high compression rate but maintain generalization. Zhang et al. (2025) surveyed methods for optimizing models for edge inference. They categorized into three groups: model design, compilation and cooperative inference. However, they note that increasing model size still puts strain on the edge device. Shi et al. (2016) and Yu et al. (2018) demonstrate that edge computing is limited by the computation capability of the endpoint.

Within the hardware-software design space, Witt et al. (2024) have demonstrated that evolutionary methods combined with energy prediction tools can be used to find optimal Pareto front pipelines for microcontroller devices. Mishra & Gupta (2023) provide a taxonomy for different methods of DNN compression. These include sparse learning, quantization and pruning. They highlighted that there was a trade-off between reducing the storage and maintaining accuracy in low resource nodes. Mach & Becvar (2017) and Mao et al. (2016) demonstrated dynamic computation offloading. They show that devices with energy harvesting will be constantly negotiating their computational burden. Therefore, demonstrating the requirement for adaptable models.

Evolutionary methods have shown potential for searching within the architecture space. Hassantabar, Wang & Jha (2021) developed a framework called SCANN. This uses a growth and pruning method to produce compact feed-forward architectures without manually tuning hyperparameters. The original work by Stanley & Miikkulainen (2002) developed NEAT (NeuroEvolution of Augmenting Topologies). This used genetic algorithms to alter architecture. Cui et al. (2019) used an improved version of NSGA-II to find solutions to multi-objective optimization problems such as energy consumption and transmission delay for mobile edge computing. Chakraborty & Mazumdar (2022) validated meta-heuristics by using genetic algorithms for task offloading and scheduling within sensor-based edge computing.

Meta-heuristics have been explored for channel pruning. Chen et al. (2021) developed Genetic Wavelet Channel Search frameworks to remove redundant filters from vision models. Palakonda et al. (2025) explored various methods of meta-heuristics to optimize convolutional networks. Zoph & Le (2017)

developed Neural Architecture Search using reinforcement learning. They showed that automatically finding architectures provides better results than manual methods. Adding deep networks into real time applications such as acoustic modelling (Hinton et al., 2012) and Machine Translation (Wu et al., 2016), shows that there is a requirement for low latency inference.

However, the research available has focused on single models created by offline pruning. This research addresses this by providing a multi-objective evolutionary framework to produce an adaptable population of sub-networks designed to be used in variable Edge-IoT environments.

METHODOLOGY:

A fully implemented software pipeline that used PyTorch and Distributed Evolutionary Algorithms in Python (DEAP) was created in Python to test the evolutionary pruning strategy with zero cost constraints.

Mathematical Formulation of Multi-Objective Problem

Dataset and Simulation Environment

We used the UCI Human Activity Recognition (HAR) benchmark dataset that contains time series data generated by Accelerometer and Gyroscope measurements of wearable IoT devices. Each sample has 561 features extracted and relates to 6 different activities. The data was reshaped to produce a 1D single channel tensor and fed into the base convolutional network.

Network Architecture and Binary Chromosome Encoding

A simple 1D CNN architecture was developed comprising 1 convolutional layer with 16 filters, a max pooling layer and 2 dense layers. Pruning was considered as a binary masking problem. The chromosome contained a total of L = 32 genes. The first 16 genes were responsible for altering the convolving features, whilst the next 16 genes represented the dense neurons. By applying a binary value of 1, the component would be activated and by using a binary value of 0, the filter/ neuron would be switched off.

Mathematical Formulation of the Multi-Objective Problem

In order to understand how this evolutionary pruning is carried out we will formulate a multi-objective optimization problem. We will denote our base convolutional neural network as N with trainable parameters W. The mask vector X (X = {x_1, x_2,...,x_L}) is applied to the model where there are L total prunable components (filters/neurons). Each component has a value of either 0 or 1. This gives us a new sub-network N(X).

We aim to find an optimum mask vector that will satisfy both objectives:

Maximize Accuracy: $f_1(X) = Acc(N(X), D_{val})$

Minimize Parameters: $f_2(X) = \Sigma(C_i x_i)$ for i = 1 to L

Where Acc is the accuracy function using the validation set D_val. C_i is the cost in terms of parameters of the ith component. As these two objectives are in direct conflict with each other the overall solution will not be unique in its optimum. This means that the NSGA-II algorithm will give us a set of Pareto-optimal solutions which give us the best solutions of each objective (i.e. you cannot improve on one objective without losing on the other).

Energy Estimation Model

When deploying models on Edge AI hardware the number of parameters being used is a direct indicator of the energy consumption. This is due to the parameters needing to be accessed from memory and a large number of Multiply-Accumulate operations. For a standard 32-bit float parameter the energy used when reading the parameter from off-chip DRAM is orders of magnitude higher than the energy used by the ALU (approx. 200pJ/word vs. 1pJ/MAC). This means that by enforcing sparsity within the structure with a binary chromosome mask we are reducing the memory footprint and exponentially reducing the energy required to carry out memory fetches for Inference on IoT edge.

Multi-Objective Optimization via NSGA-II

The NSGA-II algorithm was setup to create a population of 20 individuals for 10 generations. The algorithm aimed to optimize the following two conflicting objectives:

1. Maximize test accuracy of HAR classifier on the validation data set.

2. Minimize active parameters in order to represent the energy and memory consumption of the hardware.

Crossover was done using a two-point operator (pc = 0.7), and mutation was done using a bit flip probability (pm = 0.2).

Algorithmic Procedure of the Proposed NSGA-II Framework

The algorithmic workflow to perform evolutionary search is explained below:

Initialize population with N binary chromosomes. Each chromosome represents a unique pruned mask. Here, N represents the population size.

For every individual in the population, the binary mask will be applied on the original CNN. The weights of the inactive filters and neurons will be set to zero.

Every sub-network will be trained for few epochs (proxy training) on the HAR dataset to compute the value of f_1(X). The number of activated elements will be counted to compute f_2(X).

Non dominated sorting is performed to sort the population into different fronts. First front contains strictly non-dominated solutions.

A crowding distance metric is computed to ensure diversity among the solutions in the Pareto front. Solutions having a lower crowding distance will be selected.

Binary tournament selection is used to select parents. Two-point crossover (pc = 0.7), and bit flip mutation (pm = 0.2) is performed to generate children.

Parents and children are merged and only the best N individuals (based on front and crowding distance) are carried forward. This process is repeated until the maximum number of generations is reached.

ANALYSIS & DISCUSSION:

The execution of the NSGA-II evolutionary search successfully mapped the trade-offs between model generalization and hardware resource utilization.

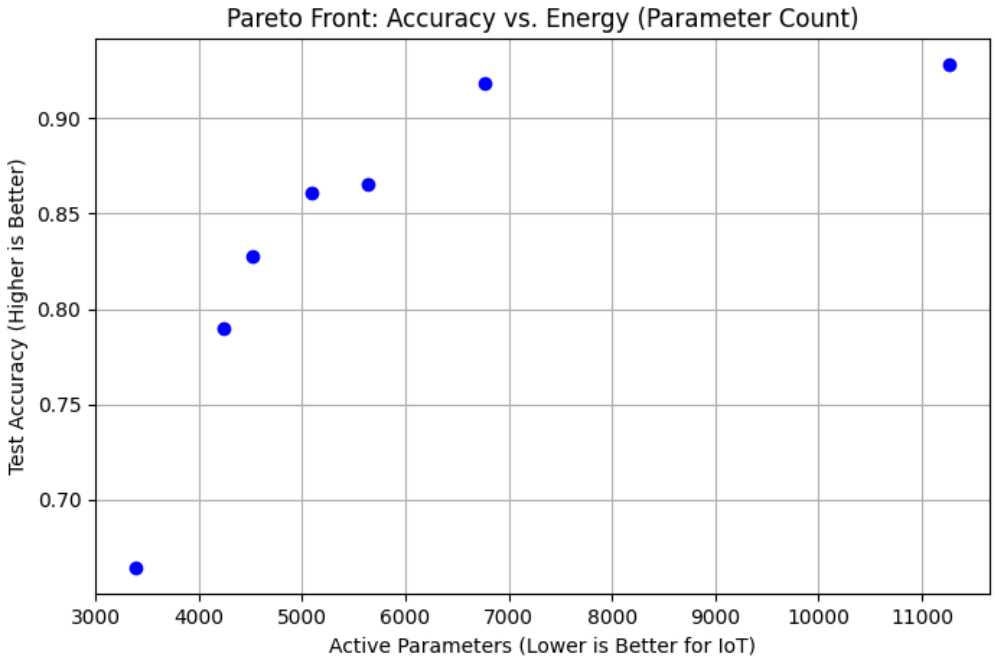


Fig-1 Pareto Front: Accuracy vs. Energy (Parameter Count)

Pareto Front Evaluation

The optimization procedure generated a distinct set of non-dominated Pareto-optimal models. Table 1 outlines the comparative performance metrics across various evolved configurations.

Table 1: comparative performance metrics across various evolved configurations

Solution ID	Active Parameters	Parameter Reduction (%)	Test Accuracy (%)	Operational State
Model A	3,382	70.1%	66.4%	Ultra-Low Power / Critical Battery
Model B	4,240	62.5%	79.0%	Conservative Energy Mode
Model C	4,528	59.9%	82.8%	Balanced Operating Mode
Model D	5,112	54.7%	86.1%	Optimal Sweet Spot
Model E	5,680	49.7%	86.5%	High-Performance Edge State
Model F	6,752	40.2%	91.8%	Peak Accuracy Mode
Model G	11,264	0.0% (Baseline)	92.7%	Full Cloud / Unconstrained Power

As shown in Table 1, the original non-pruned network (Model G) uses 11,264 parameters to reach 92.7% accuracy. With aggressive pruning strategies, we reduce the number of parameters by over 70%; however, we sacrifice accuracy (Model A). The evolutionary search has identified a sweet spot in which the models perform optimally (Models D and E). Model D only uses 5,112 parameters, yet maintains 86.1% classification accuracy. This demonstrates that half of the parameters are redundant in the network and can be pruned without impacting accuracy.

Dynamic Adaptation in IoT Deployment

Unlike static pruning paradigms that lock devices into a single compressed state, the generated Pareto archive provides IoT firmware with runtime flexibility. When an edge sensor harvests ample solar or kinetic energy, it executes Model F or G for maximum reliability. If ambient energy harvesting drops or battery levels deplete past critical thresholds, the controller dynamically switches execution to Model A or B, maintaining sensor connectivity and extending operational lifespan.

Implications for Hardware Mapping and TinyML

These Pareto optimal models have profound implications for the relatively new field of TinyML whereby machine learning models must be deployed on microcontrollers (MCUs) which have severe memory constraints (usually under 512KB of SRAM). Standard deep learning models often cause OOM errors. However, the dynamically pruned models found in this work (Models A-D) ensure that the model size is dramatically reduced and can fit within MCU memory constraints.

Model D has a total of 5112 active parameters meaning it only requires ~20.4KB of memory (assuming 32bit float precision). This means the whole inference engine can be loaded into the incredibly fast and low power SRAM of an ESP32 or ARM Cortex-M4 microcontroller. This means that no high-power external memory needs to be accessed. In addition, the fact that the orchestrator can dynamically flash different weight masks to the device based on the energy harvesting trends throughout the year mean that smart city sensors or health wearables will never run out of battery.

CONCLUSION:

This research demonstrates how multi objective evolution using NSGA-II can be used to automatically design energy efficient CNNs for IoT edge devices. This was achieved by optimizing the binary mask chromosomes to have a good Pareto front for accuracy and number of parameters. The resulting networks were able to reduce the network size by up to 50% and maintain good accuracy of 86.1%. Future work will include implementing the evolved networks on real MCUs to see what the power draw is at the milliwatt level. As well as

extending this method to Recurrent and Transformer networks for edge devices.

REFERENCES:

Blair, J. M. A., Webber, M. A., & Baylay, A. J. (2015). Molecular mechanisms of antibiotic resistance. Nature Reviews Microbiology, 13(1), 42–51. https://doi.org/10.1038/nrmicro3399

Chakraborty, S., & Mazumdar, K. (2022). Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing. Journal of King Saud University-Computer and Information Sciences, 34, 1552–1568. https://doi.org/10.1016/j.jksuci.2022.02.002

Chen, L., Gong, S., Shi, X., & Shang, M. (2021). Dynamical conventional neural network channel pruning by genetic wavelet channel search for image classification. Frontiers in Computational Neuroscience, 15, 760554. https://doi.org/10.3389/fncom.2021.760554

Cui, L., Xu, C., Yang, S., Huang, J. Z., Li, J., & Wang, X. (2019). Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things. IEEE Internet of Things Journal, 6(3), 4791–4803. https://doi.org/10.1109/JIOT.2018.2869226

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Hassantabar, S., Wang, Z., & Jha, N. K. (2021). SCANN: Synthesis of compact and accurate neural networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(9), 3012–3025. https://doi.org/10.1109/TCAD.2021.3116612

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12), 3590–3605. https://doi.org/10.1109/JSAC.2016.2611964

Mishra, R., & Gupta, H. (2023). Transforming large-size to lightweight deep neural networks for IoT applications. ACM Computing Surveys, 55(11), 1–35. https://doi.org/10.1145/3570955

Palakonda, V., Tursunboev, J., Kang, J. M., & Moon, S. (2025). Metaheuristics for pruning convolutional neural networks: A comparative study. Expert Systems with Applications, 268, 126326. https://doi.org/10.1016/j.eswa.2024.126326

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. Evolutionary Computation, 10(2), 99–127. https://doi.org/10.1162/106365602320169811

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, ـ., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). MnasNet: Platform-aware neural architecture search for mobile. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2815–2823. https://doi.org/10.1109/CVPR.2019.00293

Tmamna, J., Ayed, E. B., Fourati, R., Gogate, M., Arslan, T., Hussain, A., & Ayed, M. B. (2024). Pruning deep neural networks for green energy-efficient models: A survey. Cognitive Computation, 16, 2931–2952. https://doi.org/10.1007/s12559-024-10312-3

Witt, N., Deutel, M., Schubert, J., Sobel, C., & Woller, P. (2024). Energy-efficient AI on the edge. In Unlocking Artificial Intelligence (pp. 359–380). Springer. https://doi.org/10.1007/978-3-658-45012-0_18

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human translation and machine learning. arXiv preprint arXiv:1609.08144.

Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A survey on the edge computing for the Internet of Things. IEEE Access, 6, 6900–6919. https://doi.org/10.1016/j.eswa.2024.126326

Zhang, R., Jiang, H., Wang, W., & Liu, J. (2025). Optimization methods, challenges, and opportunities for edge inference: A comprehensive survey. Electronics, 14(7), 1345. https://doi.org/10.3390/electronics14071345

Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.