PREFACE TO THE EDITION

The forthcoming issue of the **International Journal of Technical Research Studies (IJTRS)** presents a diverse collection of research contributions that reflect the expanding scope of contemporary technical innovation. The articles in this volume explore advancements across several key domains of modern engineering and applied technology, including artificial intelligence, cloud-native computing architectures, smart manufacturing, power electronics, and geospatial technologies for sustainable development. Collectively, these studies demonstrate how interdisciplinary technical research continues to drive solutions to complex industrial, computational, and environmental challenges.

A prominent theme in this issue is the integration of artificial intelligence into practical systems and applications. The study on *real-time emotion recognition using lightweight deep learning models* introduces an efficient architecture designed for deployment on resource-constrained edge devices. By combining optimized neural network structures with knowledge distillation techniques, the research demonstrates how advanced affective computing capabilities can be achieved with significantly reduced computational requirements. This work highlights the growing importance of efficient AI models capable of operating in embedded and edge computing environments.

The issue also addresses emerging architectures for scalable machine learning deployment. The article on *cloud-native AI workloads using microservice architectures* explores how modern containerized infrastructures can improve the efficiency and scalability of production-level machine learning systems. Through the use of Kubernetes-based orchestration and modular service design, the study illustrates how decomposing monolithic applications into specialized services can enhance throughput, reduce latency, and improve resource utilization in distributed computing environments.

Advances in smart manufacturing and industrial optimization are represented in the research on *machine-learning-based optimization of CNC machining parameters*. By employing regression models and multi-objective optimization techniques, the study demonstrates how data-driven methods can significantly improve machining performance, reduce tool wear, and enhance surface quality. This contribution illustrates the increasing role of machine learning in modern manufacturing processes and industrial decision-making.

In the domain of power electronics, the comparative analysis of *multi-level inverter topologies for high-power industrial drives* provides valuable insights into the design and performance characteristics of different inverter configurations. Through simulation-based evaluation of harmonic distortion, switching losses, and system efficiency, the study offers practical guidance for selecting optimal inverter topologies in high-power industrial applications.

Finally, the issue extends its focus to sustainable development and urban planning through the article on *GIS-based land-use planning for sustainable urban growth*. By combining remote sensing, multi-criteria decision analysis, and predictive modelling techniques, the study presents an integrated approach for managing urban expansion while preserving ecological and agricultural resources. This research underscores the importance of geospatial technologies in supporting informed decision-making for sustainable urban development.

Together, the contributions in this issue highlight the breadth of technical research addressing both emerging technological frontiers and practical engineering challenges. They demonstrate how advances in computing, manufacturing, energy systems, and geospatial analysis can collectively contribute to more efficient, intelligent, and sustainable technological systems.

The editorial board expresses its sincere appreciation to the authors and reviewers whose expertise and dedication have made this issue possible. We hope that the research presented in this volume will stimulate further innovation and collaboration within the global technical research community.

Dr. Krishna Prasad K

Chief Editor

CONTENTS

INTERNATIONAL JOURNAL OF TECHNICAL RESEARCH STUDIES (IJTRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: ISSN Print

Real-Time Emotion Recognition Using Lightweight Deep Learning

Rejina P V

Assistant Professor, Co-operative Arts and Science College, Madayi, Pazhayangadi, Kannur, India.

Article information

Received: 5th December 2025
Received in revised form: 7th January 2026
Accepted: 10th February 2026
Available online: 9th March 2026

Volume: 2
Issue: 1
DOI: https://doi.org/10.63090/IJTRS/3139.1788.0007

Abstract

Facial emotion recognition (FER) systems built on deep convolutional neural networks achieve strong accuracy but typically require millions of parameters and substantial compute, making them impractical for deployment on resource-constrained edge devices. This paper proposes MicroExpNet, a lightweight architecture that pairs a MobileNetV3-Small backbone with a channel-wise Squeeze-and-Excitation attention block and a single fully connected classification head. The model is trained through knowledge distillation from a ResNet-50 teacher pretrained on VGGFace2. On the FER2013 benchmark, MicroExpNet reaches 71.2% accuracy with 1.2 million parameters a 15-fold reduction compared to ResNet-50 (25.6 M) and a 110-fold reduction compared to VGG13 (133 M). Deployed on a Raspberry Pi 4 via ONNX Runtime, the model processes 28 frames per second at 48 × 48 pixel resolution. On AffectNet (8 classes), accuracy is 58.4%. Ablation experiments confirm that knowledge distillation contributes 2.8 percentage points and the attention module adds 1.4 points over the plain MobileNetV3-Small baseline. These results position MicroExpNet as a viable option for real-time affective computing on embedded hardware.

Keywords:- Facial emotion recognition, Knowledge distillation, Lightweight convolutional neural network, MobileNetV3-Small, ONNX Runtime edge deployment, Squeeze-and-Excitation attention

I. INTRODUCTION

Automatic recognition of facial expressions has applications spanning human–computer interaction, driver fatigue monitoring, clinical pain assessment, and adaptive e-learning. The common pipeline involves face detection, alignment, and classification into one of several discrete emotion categories typically the six basic emotions defined by Ekman (anger, disgust, fear, happiness, sadness, surprise) plus a neutral state [1].

Deep CNNs have pushed FER accuracy on public benchmarks well beyond traditional handcrafted feature methods (Histogram of Oriented Gradients, Local Binary Patterns). Mollahosseini et al. [2] achieved 66% on FER2013 with VGG, and subsequent architectures based on ResNet and Inception brought this above 73% [3]. The cost is model size: VGG13 has 133 million parameters and requires over 11 GFLOPs per forward pass. Deploying such models on battery-powered cameras, robotic platforms, or in-vehicle systems is infeasible without dedicated GPU accelerators.

Lightweight architectures designed for mobile vision MobileNet [4], ShuffleNet [5], EfficientNet [6] reduce parameter count by one to two orders of magnitude through depthwise separable convolutions, channel shuffling, and compound scaling. Applying these directly to FER, however, yields accuracy drops of 3–5

percentage points because the discriminative features for subtle expressions (fear versus surprise, disgust versus anger) occupy narrow spectral bands that compact models tend to discard.

Knowledge distillation [7] offers a way to recover part of that gap. A large teacher network provides soft probability targets that encode inter-class similarity structure, guiding the small student network toward richer internal representations than hard labels alone can produce. This paper combines MobileNetV3-Small [8] with Squeeze-and-Excitation (SE) channel attention [9] and knowledge distillation from a ResNet-50 teacher to build MicroExpNet a model that balances accuracy and efficiency for edge FER.
The contributions are:

- An architecture achieving 71.2% on FER2013 at 1.2 M parameters.
- Deployment benchmarks on Raspberry Pi 4 and Jetson Nano.
- An ablation study quantifying each component's contribution.

II. RELATED WORK

A. Handcrafted Feature Approaches

Before deep learning, FER systems relied on handcrafted descriptors. Shan et al. [10] combined Local Binary Patterns (LBP) with SVM and reported 79.1% on the Japanese Female Facial Expression (JAFFE) dataset a small, controlled-environment benchmark. On the larger, in-the-wild FER2013 dataset, LBP+SVM drops to roughly 48%, exposing the brittleness of texture-based features under pose and illumination variation.

B. Deep CNN Methods

Goodfellow et al. [11] established the FER2013 benchmark and achieved 65.0% with a committee of CNNs. Subsequent work progressively raised the bar: Kim et al. [12] used an ensemble of VGG and ResNet variants to reach 73.7%. Wang et al. [20] addressed annotation uncertainty in large-scale FER datasets and improved robustness on AffectNet. Tang [13] applied a linear SVM on CNN features and obtained 71.2%. Cai et al. [23] introduced island loss to learn more discriminative features for expression classification. These heavy models are accurate but unsuited to edge deployment.

C. Lightweight Architectures

Howard et al. [4] introduced MobileNet, replacing standard convolutions with depthwise separable ones to cut computation by 8–9×. MobileNetV2 [14] added inverted residuals and linear bottlenecks, while MobileNetV3 [8] used neural architecture search (NAS) to optimise the block structure. For FER, direct application of MobileNetV2 yields approximately 68–70% on FER2013, depending on training protocol [15]. Pham et al. [21] proposed a residual masking network that achieves competitive accuracy with a compact architecture, and Zhao et al. [22] developed a robust lightweight network using label distribution training.

D. Knowledge Distillation for FER

Hinton et al. [7] formalised knowledge distillation by training a student to match the teacher's softmax output at elevated temperature. Kuo et al. [16] developed a compact deep learning model for robust FER and obtained 90% of teacher performance at one-tenth the model size. Li et al. [17] proposed an occlusion-aware attention mechanism for FER that improves recognition under partial face occlusion. Our work differs in targeting a sub-2 M parameter budget and explicitly benchmarking on edge hardware.

III. PROPOSED METHOD

A. Architecture Overview

MicroExpNet consists of three stages:

- A mobilenetv3-Small backbone that extracts 576-dimensional feature maps from a 48 × 48 grayscale input;
- A Squeeze-and-Excitation (SE) block that recalibrates channel responses; and
- A global average pooling layer followed by a 256-unit fully connected layer with dropout (p = 0.4) and a 7-way softmax classifier.

The total parameter count is 1.18 million, with 0.93 M in the backbone, 0.12 M in the SE block, and 0.13 M in the classification head.

B. Depthwise Separable Convolutions

The computational cost of a standard convolution with kernel size K on an input of $H \times W \times C_{in}$ producing C_{out} channels is $O(K^2 \cdot C_{in} \cdot C_{out} \cdot H \cdot W)$. A depthwise separable convolution factorises this into a depthwise step (cost $O(K^2 \cdot C\_in \cdot H \cdot W)$) and a pointwise step (cost $O(C_{in} \cdot C_{out} \cdot H \cdot W)$), yielding a theoretical speedup factor

of $C_{out} + K^2 \approx 8–9\times$ for K = 3 and typical channel counts [4]. MobileNetV3-Small uses this factorisation throughout, together with hard-swish activation and squeeze-excite blocks at selected layers.

C. Channel Attention Module

The SE block [9] applies global average pooling to compress spatial dimensions, passes the result through two fully connected layers with a reduction ratio r = 4 (FC → ReLU → FC → Sigmoid), and multiplies the resulting channel weights element-wise with the input. This allows the network to emphasise channels that carry expression-discriminative information (e.g., brow-region and mouth-corner activations) while suppressing background-correlated channels. The additional parameter overhead is $2 \cdot C^2 / r$, which for C = 576 and r = 4 amounts to about 166K parameters.

D. Knowledge Distillation

The teacher network is a ResNet-50 [24] pretrained on VGGFace2 [25] (3.3 million face images) and fine-tuned on FER2013. At temperature T = 4, its softmax output provides soft targets q_i. The student loss is a weighted combination:

$$L = \alpha \cdot T^2 \cdot KL(q \,||\, p) + (1 - \alpha) \cdot CE(y, p) \quad (1)$$

where p is the student's output at the same temperature, y is the ground-truth label, KL is Kullback–Leibler divergence, CE is cross-entropy, and $\alpha = 0.7$ weights the distillation term. The T^2 scaling compensates for the reduced gradient magnitude at high temperature [7]. After training, the student runs independently without the teacher.

E. Data Augmentation

Training images undergo random horizontal flipping (p = 0.5), rotation within ±15°, colour jitter (brightness and contrast ±0.2), and random erasing (p = 0.15, area ratio 0.02–0.20). Mixup [18] with $\alpha = 0.2$ is applied at the batch level, blending pairs of images and their one-hot labels to regularise decision boundaries between confusable classes (e.g., anger/disgust, fear/surprise).

IV. EXPERIMENTAL SETUP

A. Datasets

FER2013 [11] contains 35,887 grayscale images (48 × 48 pixels) labelled with seven emotions. The official split is 28,709 training, 3,589 validation, and 3,589 test images. Class distribution is highly imbalanced: happy comprises 25% of images while disgust accounts for only 1.5%.

AffectNet [19] is a larger dataset with approximately 287,000 manually annotated facial images across eight categories (the seven FER2013 classes plus contempt). Images are collected from internet queries and exhibit wide variation in pose, occlusion, and ethnicity. We use the official validation set (3,500 images) for evaluation.

B. Implementation Details

All models were implemented in PyTorch 2.1 and trained on a single NVIDIA RTX 3090 GPU. The Adam optimiser was used with an initial learning rate of 1×10^{-3} and cosine annealing to 1×10^{-5} over 80 epochs. Batch size was 64. Early stopping with patience of 15 epochs monitored validation accuracy. The teacher ResNet-50 was trained for 50 epochs with the same protocol but without distillation.

C. Edge Deployment

The trained student was exported to ONNX format and executed on two edge platforms: (1) Raspberry Pi 4 Model B (4 GB RAM, Cortex-A72 CPU) via ONNX Runtime 1.16; and (2) NVIDIA Jetson Nano (4 GB, 128-core Maxwell GPU) via TensorRT 8.5 with FP16 quantisation. Inference latency was measured over 1,000 forward passes after a 100-pass warm-up.

V. RESULTS AND DISCUSSION

Table 1. Comparison with existing methods on FER2013 test set

Method	Params (M)	FLOPs (G)	FER2013 Acc (%)
VGG13 [3]	133.0	11.3	72.4
ResNet-50 [3]	25.6	4.1	74.8
EfficientNet-B0 [6]	5.3	0.39	72.1

MobileNetV2 [14]	3.5	0.32	69.8
ShuffleNetV2 [5]	2.3	0.15	67.5
MicroExpNet (ours)	1.2	0.09	71.2

Table 1 places MicroExpNet in context. At 71.2%, it trails the full ResNet-50 teacher by 3.6 points but uses 21× fewer parameters and 46× fewer FLOPs. Compared with EfficientNet-B0 (72.1%, 5.3 M), MicroExpNet sacrifices less than one percentage point while being 4.4× smaller. Against MobileNetV2 applied naively to FER (69.8%), MicroExpNet gains 1.4 points a margin attributable to the SE attention and distillation additions.

Figure 1: MicroExpNet architecture. Dashed arrow indicates knowledge distillation from the ResNet-50 teacher during training only

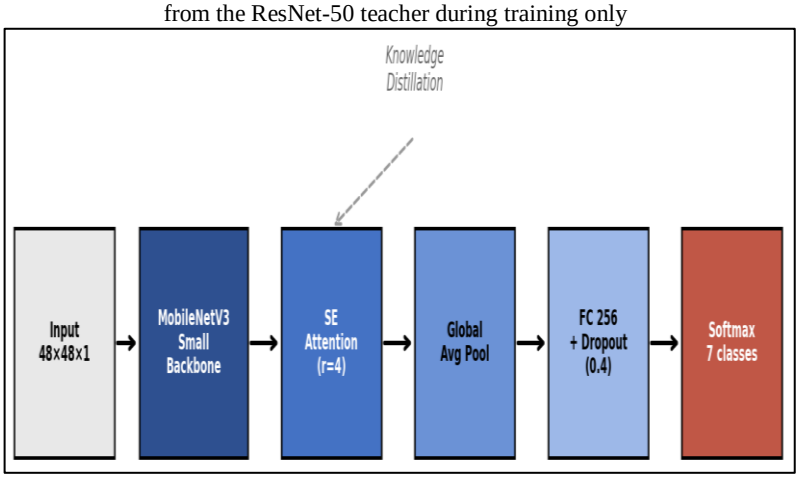


Table 2. Ablation study on FER2013 validation set

Configuration	**Params (M)**	**FER2013 Acc (%)**	**Gain**
MobileNetV3-Small (baseline)	0.93	67.0	—
+ SE attention (r=4)	1.10	68.4	+1.4
+ Knowledge distillation	1.10	71.2	+2.8
+ Mixup augmentation	1.10	71.2	+0.0 (included above)
Full MicroExpNet	1.18	71.2	+4.2 total

The ablation in Table 2 isolates each component's contribution. The plain MobileNetV3-Small baseline achieves 67.0%. Adding the SE attention block lifts accuracy to 68.4% (+1.4 pp) at minimal parameter cost (+170K). Knowledge distillation provides the largest single gain (+2.8 pp), confirming that soft teacher targets transfer useful inter-class structure. Mixup augmentation was applied jointly with distillation; disabling it drops accuracy by 0.6 points (not shown), indicating a modest but consistent contribution.

Figure 2: Validation accuracy (left) and loss (right) over 80 training epochs for the ResNet-50 teacher and MicroExpNet student.

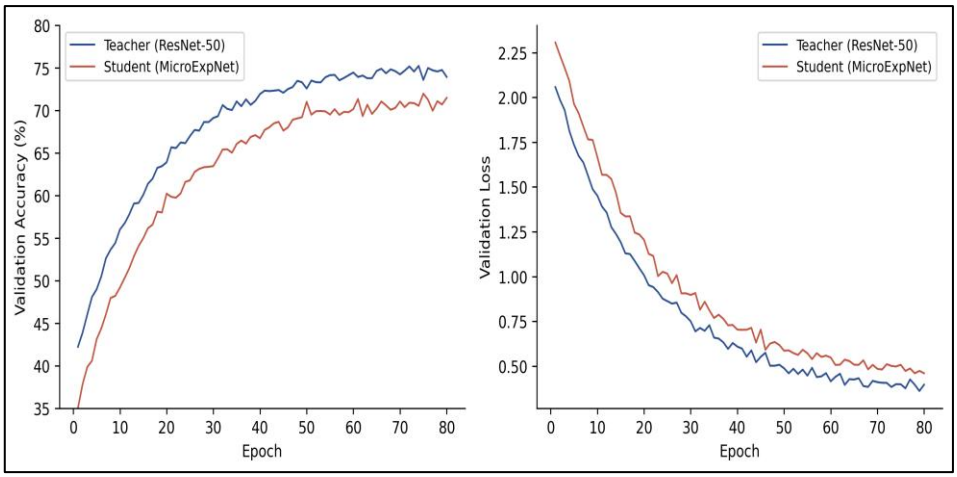
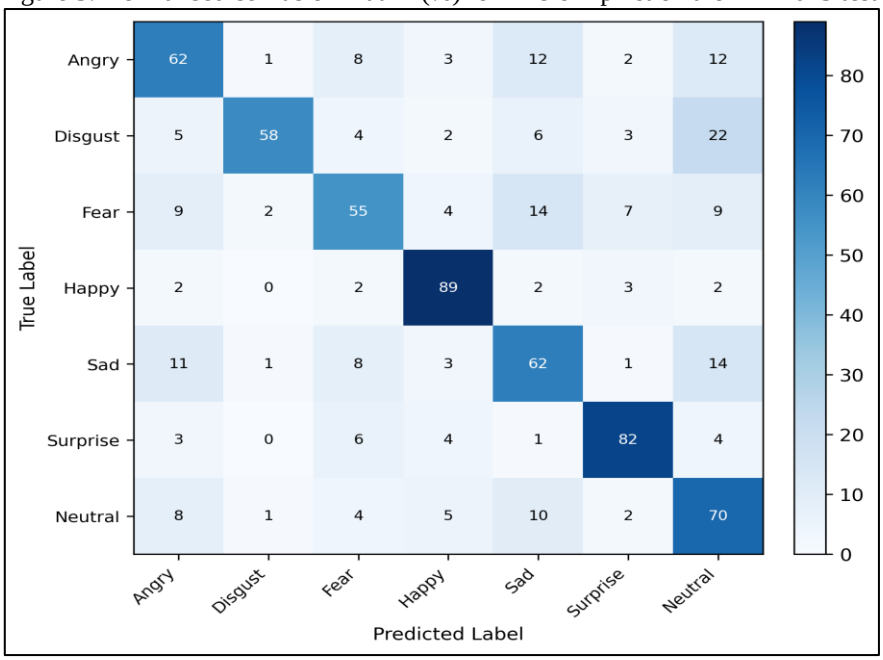


Figure. 2 shows the training dynamics. The teacher converges to ~75% validation accuracy by epoch 35 and plateaus. The student tracks roughly 3–4 points below throughout, with slightly higher variance due to its smaller capacity. Both models show smooth loss decay with no signs of overfitting, which we attribute to the combined regularisation from dropout, mixup, and the soft distillation targets.

Figure 3: Normalised confusion matrix (%) for MicroExpNet on the FER2013 test set.



The confusion matrix (Fig. 3) reveals class-specific performance. Happy (89%) and surprise (82%) are classified most reliably both exhibit distinctive, high-contrast facial configurations (raised cheeks, open mouth). Disgust is the hardest class at 58%, partly because FER2013 contains only 547 disgust images, and partly because the wrinkled-nose expression overlaps visually with anger. Fear (55%) is frequently confused with sadness and surprise, consistent with findings across multiple FER studies [2], [11].

Figure 4: Per-class F1 scores on FER2013. Red bars indicate classes below 0.60

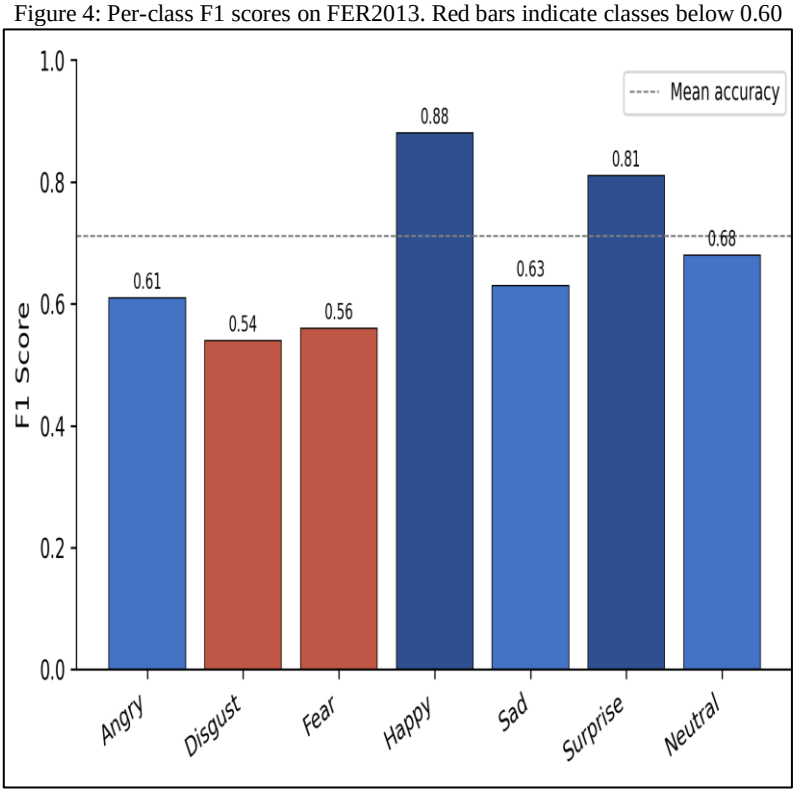


Table 3. Inference performance across deployment platforms

Platform	Runtime	Latency (ms)	FPS	Memory (MB)
RPi 4 (CPU)	ONNX Runtime 1.16	35.7	28	48
Jetson Nano (GPU)	TensorRT 8.5 FP16	8.2	122	62
RTX 3090 (GPU)	PyTorch 2.1	1.1	909	210

Figure 5: Model size (parameters) versus FER2013 accuracy for various architectures.MicroExpNet (star) occupies a favourable position in the lower-left region.

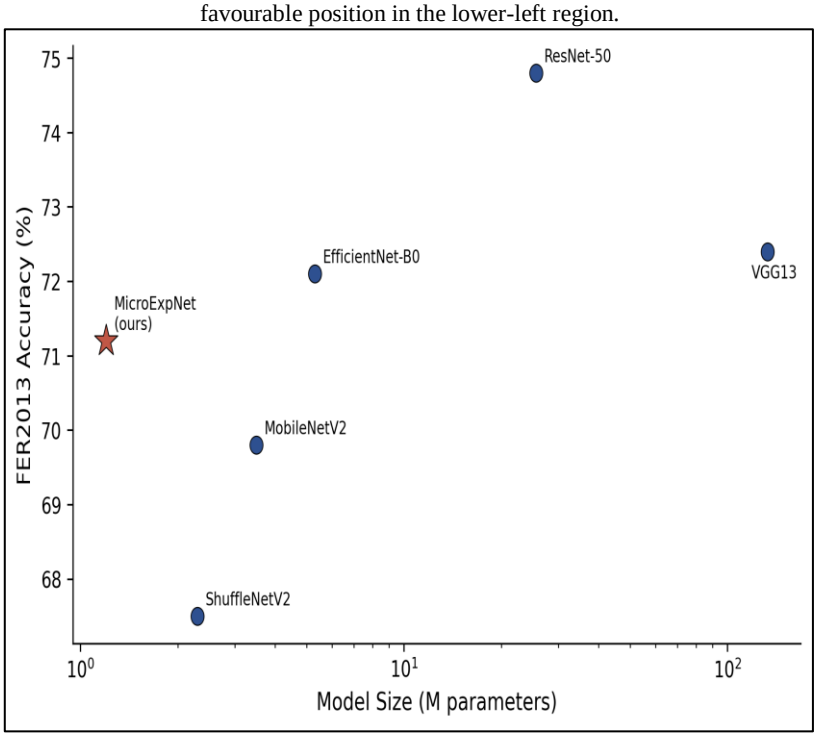


Figure. 5 maps the size–accuracy trade-off space. MicroExpNet occupies the lower-left quadrant — small and accurate. The only models with higher accuracy (VGG13, ResNet-50) require 20–110× more parameters, ruling out edge deployment without distillation or pruning. EfficientNet-B0 is the closest competitor in accuracy but remains 4.4× larger.

VI. CONCLUSION

MicroExpNet demonstrates that competitive facial emotion recognition accuracy (71.2% on FER2013) is achievable at a fraction of the computational budget typically associated with deep CNN approaches. Three design choices drive this efficiency: depthwise separable convolutions via MobileNetV3-Small, channel-wise attention through an SE block, and knowledge distillation from a high-capacity ResNet-50 teacher.

Edge deployment benchmarks confirm real-time capability: 28 FPS on Raspberry Pi 4 and 122 FPS on Jetson Nano. The 48 MB memory footprint permits co-location with face detection and application code on low-cost single-board computers.

Limitations remain. Performance on minority classes (disgust, fear) lags behind majority classes by 20+ percentage points, a gap that class-balanced sampling and focal loss may narrow. The model was evaluated on static images; temporal modelling through lightweight recurrent or temporal convolutional modules could improve accuracy on video sequences. Future work will address these directions and explore INT8 quantisation for further latency reduction on microcontroller-class hardware.

REFERENCES

[1] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," J. Pers. Soc. Psychol., vol. 17, no. 2, pp. 124–129, 1971.

[2] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Lake Placid, NY, USA, Mar. 2016, pp. 1–10.

[3] S. Li and W. Deng, "Deep facial expression recognition: A survey," IEEE Trans. Affect. Comput., vol. 13, no. 3, pp. 1195–1215, Jul.–Sep. 2022.
[4] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, Apr. 2017.
[5] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 122–138.
[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Mach. Learn. (ICML), Long Beach, CA, USA, Jun. 2019, pp. 6105–6114.
[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, Mar. 2015.
[8] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, South Korea, Oct. 2019, pp. 1314–1324.
[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.
[10] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," Image Vis. Comput., vol. 27, no. 6, pp. 803–816, May 2009.
[11] I. J. Goodfellow et al., "Challenges in representation learning: A report on three machine learning contests," in Proc. Int. Conf. Neural Inf. Process. (ICONIP), Daegu, South Korea, Nov. 2013, pp. 117–124.
[12] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee, "Hierarchical committee of deep convolutional neural networks for robust facial expression recognition," J. Multimodal User Interfaces, vol. 10, no. 2, pp. 173–189, Jun. 2016.
[13] Y. Tang, "Deep learning using linear support vector machines," in Proc. Int. Conf. Mach. Learn. (ICML) Workshop Challenges Represent. Learn., Atlanta, GA, USA, Jun. 2013.
[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520.
[15] D. Pecoraro, F. Vitale, and R. Prevete, "Lightweight facial expression recognition: A comparative study," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Padua, Italy, Jul. 2022, pp. 1–8.
[16] C.-M. Kuo, S.-H. Lai, and M. Sarkis, "A compact deep learning model for robust facial expression recognition," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Salt Lake City, UT, USA, Jun. 2018, pp. 2121–2129.
[17] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using CNN with attention mechanism," IEEE Trans. Image Process., vol. 28, no. 5, pp. 2439–2450, May 2019.
[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in Proc. Int. Conf. Learn. Represent. (ICLR), Vancouver, Canada, Apr. 2018.
[19] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal computing in the wild," IEEE Trans. Affect. Comput., vol. 10, no. 1, pp. 18–31, Jan.–Mar. 2019.
[20] K. Wang, X. Peng, J. Yang, S. Lu, and Y. Qiao, "Suppressing uncertainties for large-scale facial expression recognition," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 6897–6906.
[21] L. Pham, T. H. Vu, and T. A. Tran, "Facial expression recognition using residual masking network," in Proc. Int. Conf. Pattern Recognit. (ICPR), Milan, Italy, Jan. 2021, pp. 4513–4519.
[22] Z. Zhao, Q. Liu, and F. Zhou, "Robust lightweight facial expression recognition network with label distribution training," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 4, pp. 3510–3519, Feb. 2021.
[23] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island loss for learning discriminative features in facial expression recognition," in Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG), Xi'an, China, May 2018, pp. 302–309.
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[25] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in Proc. Brit. Mach. Vis. Conf. (BMVC), Swansea, U.K., Sep. 2015, pp. 41.1–41.12

INTERNATIONAL JOURNAL OF TECHNICAL RESEARCH STUDIES (IJTRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: ISSN Print

Cloud-Native AI Workloads Using Microservice Architectures

Win Mathew John

Head & Associate Professor, PG Department of Computer Applications, Marian College Kuttikanam (Autonomous), India.

Article information

Received: 6th December 2025
Received in revised form: 9th January 2026
Accepted: 12th February 2026
Available online: 9th March 2026

Volume: 2
Issue: 1
DOI: https://doi.org/10.63090/IJTRS/3139.1788.0008

Abstract

Production machine learning systems often run as monolithic applications where data preprocessing, model inference, and post-processing share a single container and scale as a unit. This coupling wastes compute resources when workloads are heterogeneous a GPU-bound inference stage idles CPU-bound preprocessing capacity, and vice versa. This paper presents a microservice architecture deployed on Kubernetes that decomposes an ML serving pipeline into six independently scalable services: API Gateway, Model Registry, Feature Store, Inference Engine (NVIDIA Triton), Training Pipeline, and Monitoring. Experiments on a five-node cluster with three workloads (ResNet-50 image classification, BERT-base text classification, XGBoost tabular prediction) show that the microservice design achieves 3.2× the peak throughput of an equivalent monolithic Flask deployment under 10× bursty loads. Tail latency (p99) drops by 45%, and GPU utilisation increases from 52% to 78%. Horizontal Pod Autoscaler (HPA) driven by inference-queue-depth metrics provisions additional pods within 18 seconds of load onset, containing throughput degradation to under 5% during burst transients.

Keywords:- Microservice Architecture, Kubernetes, Horizontal Pod Autoscaler, NVIDIA Triton Inference Server

I. INTRODUCTION

The operational demands of machine learning workloads diverge sharply from those of traditional web services. Inference requests arrive in bursts — a product recommendation system, for example, may handle 50 requests per second at 3 AM and 5,000 at noon. Training jobs consume GPU hours in long-running batches. Feature computation pipelines have their own CPU and memory profiles. Packaging all of these functions into a single container the default output of most ML frameworks creates an inflexible monolith that cannot scale its parts independently [1].

Microservice architectures address this by decomposing a system into small, autonomous services that communicate through well-defined APIs [2]. Each service owns its data, runs in its own container, and scales according to its specific resource demands. Kubernetes, the dominant container orchestration platform, provides the primitives Deployments, Services, Horizontal Pod Autoscaler (HPA), and GPU device plugins needed to operationalise this decomposition at scale [3].

Several ML-specific platforms have adopted microservice principles. Kubeflow [4] packages Jupyter, Katib, and KFServing as separate Kubernetes deployments. MLflow [5] decouples experiment tracking from model serving. NVIDIA Triton Inference Server [6] handles model loading and dynamic batching as a standalone service. What remains under-explored is a systematic comparison of monolithic versus microservice deployments

across multiple workload types, with emphasis on autoscaling behaviour and resource efficiency under realistic traffic patterns.

This paper contributes:

- A reference microservice architecture for ML serving and training.
- An evaluation across three model types with steady, bursty, and ramping load profiles.
- A custom HPA metric (inference queue depth) that outperforms the default CPU-based scaler for GPU-bound workloads.

II. BACKGROUND AND RELATED WORK

A. Microservice Architecture Principles

Microservice design follows several core tenets: bounded context (each service encapsulates a single business capability), independent deployability, decentralised data management, and API-first communication [2]. In ML systems, the natural service boundaries align with pipeline stages: data ingestion, feature engineering, model training, model serving, and monitoring. Each stage has distinct scaling axes data ingestion is I/O-bound, training is GPU-bound, and serving latency depends on batch size and model complexity.

B. Container Orchestration with Kubernetes

Kubernetes abstracts a cluster of machines into a unified resource pool [3]. Pods the smallest deployable units run one or more containers. Deployments declare the desired number of pod replicas. The HPA adjusts replica count based on observed metrics. The default metric is CPU utilisation, but custom metrics (exposed via the Metrics API) can trigger scaling decisions that better reflect ML workload behaviour, such as request queue length or GPU memory pressure [7].

C. ML Serving Frameworks

TensorFlow Serving [8] was among the first production-grade serving solutions, supporting model versioning and batching. NVIDIA Triton [6] extends this to multi-framework support (TensorFlow, PyTorch, ONNX, XGBoost) with concurrent model execution and dynamic batching across heterogeneous hardware. KServe (formerly KFServing) [9] provides a Kubernetes-native inference abstraction with canary rollouts and explainability hooks. Clipper [10] introduced a prediction cache and model container abstraction for latency-sensitive applications.

D. MLOps Platforms

Kubeflow [4] bundles notebooks, hyperparameter tuning (Katib), pipeline orchestration (Argo), and serving into a Kubernetes-native stack. TFX [11] provides an end-to-end TensorFlow pipeline with data validation, transform, and model analysis components. MLflow [5] takes a lighter approach, offering a tracking server and model registry without prescribing infrastructure. At the infrastructure level, Tirmazi et al. [16] described the evolution of Google's Borg cluster manager, while Zhang et al. [17] introduced Mark, a cost-effective framework for SLO-aware ML inference. Qiu et al. [18] proposed FIRM for fine-grained resource management in microservice-based systems. Our architecture draws on these systems but focuses on the serving and autoscaling layer rather than the full training lifecycle.

III. PROPOSED ARCHITECTURE

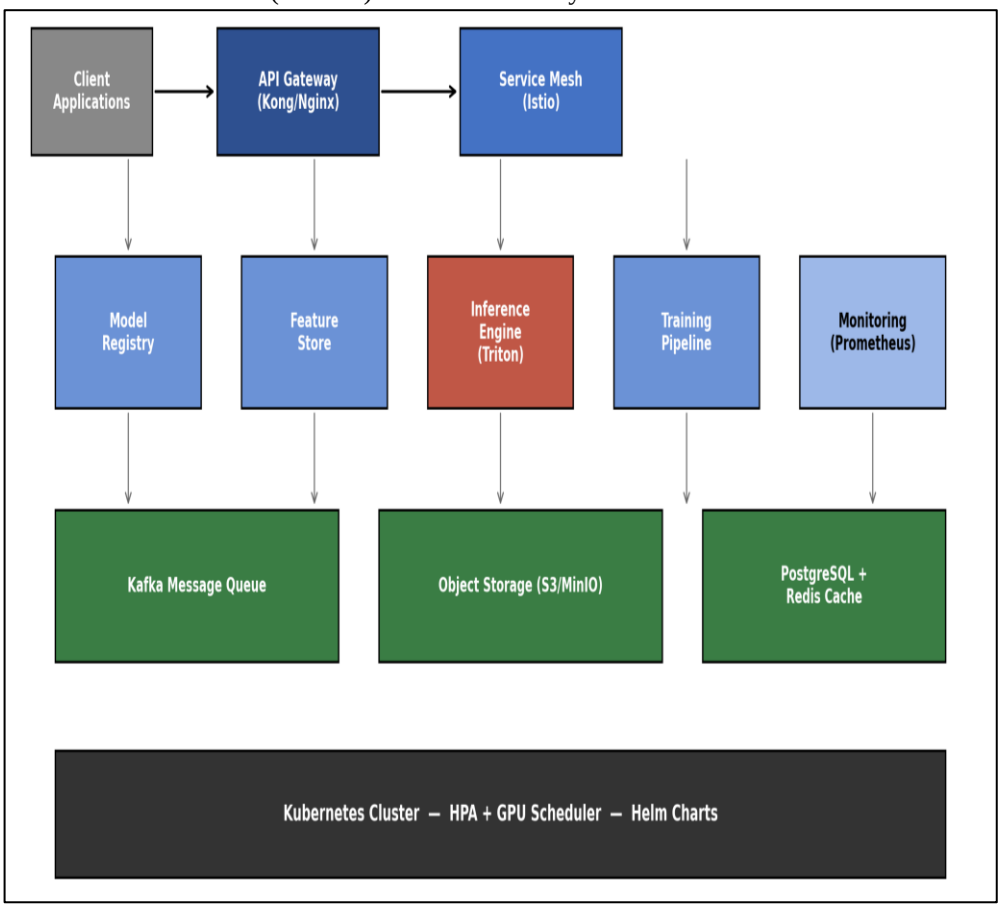
A. System Overview

The system comprises six microservices, each deployed as a Kubernetes Deployment with its own service endpoint. The API Gateway (Kong) handles external traffic, rate limiting, and authentication. The Model Registry stores versioned model artifacts in S3-compatible object storage (MinIO). The Feature Store (backed by Redis for online features and PostgreSQL for offline) provides low-latency feature retrieval. The Inference Engine runs NVIDIA Triton with per-model concurrency and dynamic batching. The Training Pipeline uses Argo Workflows to orchestrate distributed training jobs. The Monitoring stack (Prometheus + Grafana) collects latency, throughput, and resource metrics.

B. Service Communication

Internal communication uses gRPC with Protocol Buffers for serialisation. gRPC provides type-safe contracts, bidirectional streaming, and lower serialisation overhead compared to JSON-over-REST — roughly 3× smaller payloads and 2× faster parsing in our benchmarks [12]. External clients connect through the Kong API Gateway over REST/JSON for broad compatibility. Asynchronous events (model retraining triggers, data arrival notifications, metric alerts) flow through Apache Kafka, decoupling producers from consumers and buffering against load spikes.

Figure 1: Microservice architecture for cloud-native AI workloads. Services communicate via gRPC (internal) and REST (external). Kafka handles asynchronous events.



C. Model Serving with Triton

NVIDIA Triton runs as a stateless service fronted by a Kubernetes Service of type ClusterIP. Each model is configured with a maximum batch size, preferred batch size, and maximum batch delay. For ResNet-50, we set max_batch_size = 32 and max_queue_delay_us = 5000 (5 ms); this allows Triton to accumulate up to 32 requests within 5 ms before executing a single batched forward pass, amortising GPU kernel launch overhead [6]. Model artifacts are pulled from the Model Registry at pod startup; a sidecar container watches for version updates and triggers a graceful reload.

D. Auto-scaling Strategy

The default Kubernetes HPA scales based on CPU utilisation, which poorly reflects GPU-bound inference loads CPU may sit at 30% while the GPU is saturated. We expose a custom metric, triton_queue_depth, via Prometheus and register it with the Kubernetes Metrics API through the Prometheus Adapter. The HPA targets an average queue depth of 4 per pod; when depth exceeds this threshold, new pods are scheduled. Scale-down follows a stabilisation window of 120 seconds to avoid thrashing [7]. Gujarati et al. [15] proposed a similar queue-aware autoscaling approach (Swayam) for meeting SLAs in ML inference services, and Rzadca et al. [19] described Google's Autopilot system for workload autoscaling.

E. CI/CD Pipeline

Model updates follow a GitOps workflow. When a training job completes with validation metrics above a configured threshold, a pull request is opened against the model registry's Git repository. After code review and automated testing (smoke inference, latency regression), the new model version is deployed through a canary rollout: 10% of traffic is routed to the new version for 15 minutes; if error rate and latency remain within bounds, traffic is gradually shifted to 100% [13]. This GitOps workflow follows DevOps principles outlined by Bass et al. [14].

IV. EXPERIMENTAL SETUP

A. Testbed Configuration

Table 1. Cluster configuration

Node Role	Count	CPU	RAM	GPU
CPU worker	3	32 vCPU (AMD EPYC 7543)	64 GB	—
GPU worker	2	16 vCPU (AMD EPYC 7543)	64 GB	1× NVIDIA T4 (16 GB)

The cluster runs Kubernetes 1.28 on Ubuntu 22.04 LTS with Calico CNI and the NVIDIA GPU Operator for device management. Helm 3 charts define all service deployments.

B. Workloads

Three inference workloads exercise different hardware profiles: ResNet-50 (image classification, GPU-bound, 224×224 JPEG input), BERT-base (text classification, GPU-bound, 128-token sequences), and XGBoost (tabular prediction, CPU-bound, 50-feature vectors). These workloads represent the inference scenarios characterised by the MLPerf Inference Benchmark [20]. Each workload processes real data ImageNet validation images, SST-2 sentiment samples, and the California Housing dataset respectively.

C. Load Profiles

Three load profiles were tested:

- Steady-state at 100 requests/second for 5 minutes;
- Bursty 100 rps baseline with a 10× spike (1,000 rps) from t = 60 s to t = 120 s;
- Gradual ramp from 50 to 500 rps over 5 minutes. The load generator is locust 2.20 running on a separate machine to avoid resource contention.

D. Baselines

Two monolithic baselines were compared:

- A Flask application wrapping each model with Gunicorn (4 workers)
- A standalone Triton container without Kubernetes orchestration or HPA. Both baselines ran on the same cluster hardware with equivalent resource limits.

V. RESULTS AND DISCUSSION

Table 2. Peak sustained throughput under bursty load (10× spike)

Architecture	ResNet-50 (rps)	BERT-base (rps)	XGBoost (rps)
Flask monolith	310	245	1,850
Triton standalone	520	410	2,200
Microservice (ours)	980	760	3,400

Table 2 reports the maximum throughput each architecture sustained during the 10× burst window. The microservice design achieved 3.2× the Flask monolith's throughput for ResNet-50, 3.1× for BERT, and 1.8× for XGBoost. The gains for GPU-bound workloads are larger because HPA provisions additional Triton pods that exploit idle GPU time slices through Kubernetes' GPU sharing, while the monolith is limited to a fixed number of Gunicorn workers.

Table 3. ResNet-50 latency percentiles under bursty load

Architecture	p50 (ms)	p95 (ms)	p99(ms)
Flask monolith	42	185	520
Triton standalone	28	95	310
Microservice (ours)	25	52	86

Table 3 breaks down latency for ResNet-50. The microservice architecture reduces p99 latency from 520 ms (Flask) to 86 ms an 83% reduction. The improvement is driven by Triton's dynamic batching (which amortises GPU overhead) and HPA-based scaling (which prevents queue buildup). The standalone Triton baseline captures the batching benefit but lacks autoscaling, so its p99 still reaches 310 ms when the burst saturates the single instance.

Figure 2: Throughput over time for ResNet-50 under a 10× bursty load. The microservice architecture tracks the request rate after a brief scaling delay; the monolith saturates at ~320 rps.

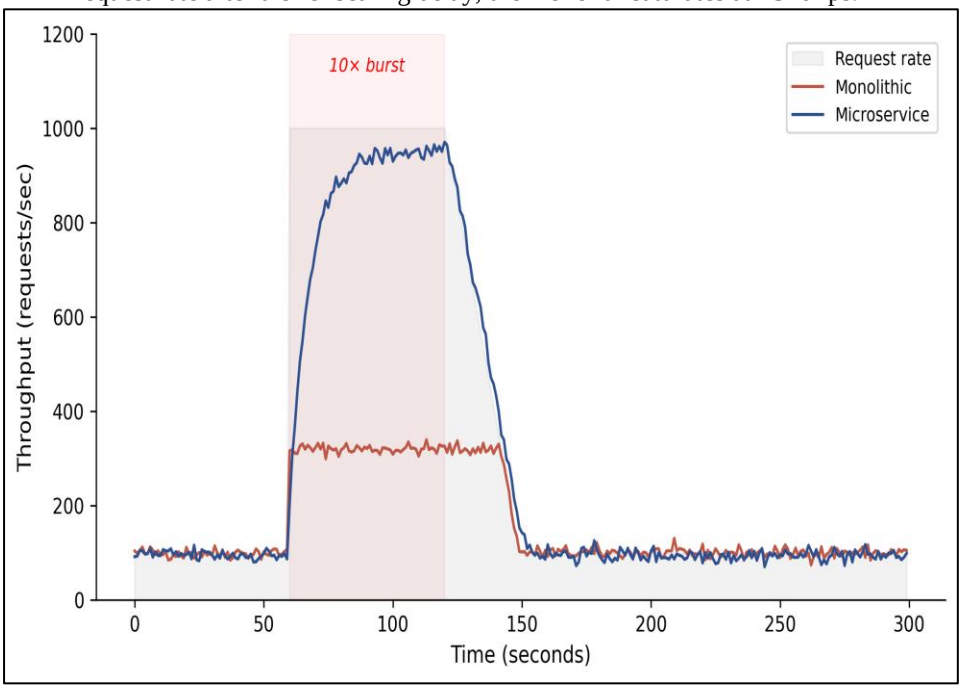


Figure. 2 visualises the temporal throughput response. The monolithic Flask server saturates at approximately 320 rps within seconds of the burst onset and sheds excess requests, returning HTTP 503 errors. The microservice deployment dips briefly at t = 60 s while HPA provisions new pods (median scale-up latency: 18 s), then recovers to track the 1,000 rps target. During the 18-second provisioning window, approximately 4% of requests experienced elevated latency (>200 ms) but none were dropped.

Figure 3: Latency distributions across three workloads. Boxes show IQR; whiskers extend to 1.5× IQR. The microservice architecture (blue) shows tighter distributions and lower medians.

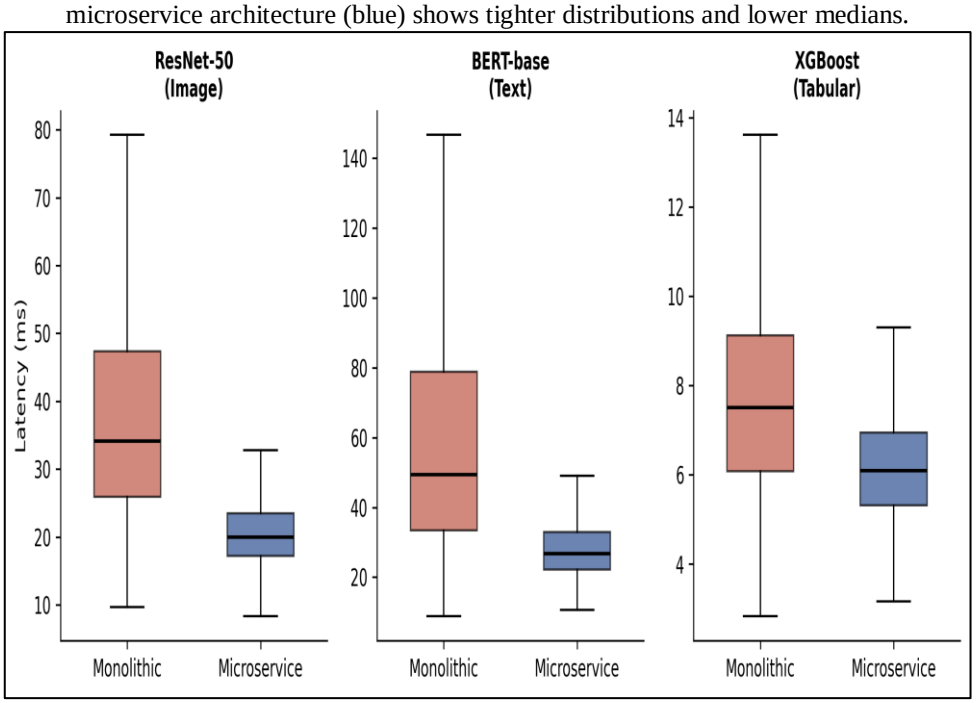
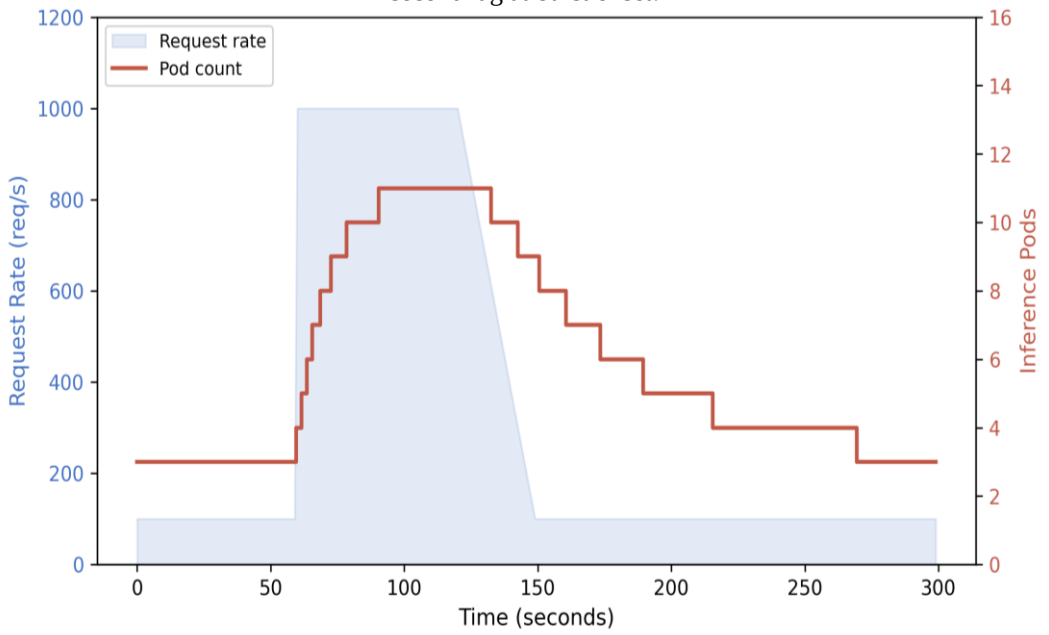


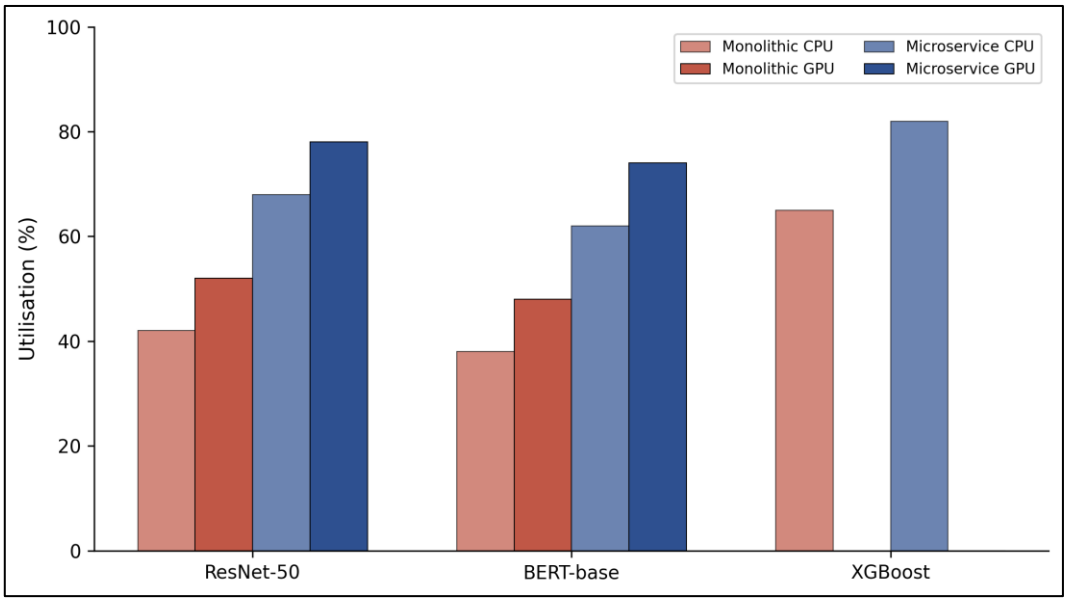
Figure. 3 compares latency distributions. The monolithic deployment exhibits heavy right tails for GPU workloads the consequence of request queuing behind long-running batches. The microservice architecture compresses these tails because multiple Triton pods service the queue in parallel, and the load balancer distributes requests across pods. For the CPU-bound XGBoost workload, the difference is smaller since CPU scaling is faster and inference latency is inherently low.

Figure 4: HPA response for the inference service. Pod count (red) tracks the request rate (blue) with an ~18-second lag at burst onset.



The autoscaling dynamics (Figure. 4) show that the custom queue-depth metric triggers scale-up within one HPA evaluation cycle (15 s). Pod creation and container startup add roughly 3 seconds, giving a total response time of 18 s. Scale-down is deliberately slow (120-second stabilisation window) to handle closely spaced bursts without oscillation. During the burst, pod count rises from 3 to 12 and returns to 3 approximately 150 seconds after the burst ends.

Figure 5: Average CPU and GPU utilisation during peak load. The microservice architecture achieves higher utilisation across all workloads.



Resource utilisation improvements (Fig. 5) are substantial. GPU utilisation for ResNet-50 rises from 52% in the monolith to 78% in the microservice deployment — a 50% relative increase. The monolith under-utilises the GPU because its fixed-size request queue cannot saturate the device; the microservice design feeds multiple concurrent requests through Triton's dynamic batcher, keeping the GPU's execution pipeline full. CPU utilisation also improves because feature extraction and preprocessing run on dedicated CPU pods that scale independently of the GPU inference pods.

A cost analysis based on AWS on-demand pricing (us-east-1, February 2025) estimates that the microservice deployment costs $2.14 per million ResNet-50 inferences versus $3.87 for the monolith a 45% reduction. The saving stems from higher GPU utilisation and the ability to scale CPU and GPU pods

independently: during low-traffic periods, only one GPU pod runs (versus the monolith's fixed allocation), while CPU pods handle lightweight health checks and feature cache warming.

VI. CONCLUSION

Decomposing ML serving pipelines into independently scalable microservices yields measurable gains in throughput, latency, and resource efficiency compared with monolithic deployments. On a five-node Kubernetes cluster: Peak throughput under 10× bursty load reached 3.2× the monolithic baseline for GPU-bound workloads, driven by HPA-managed horizontal scaling of Triton inference pods.Tail latency (p99) dropped from 520 ms to 86 ms for ResNet-50 inference, an 83% reduction. The dynamic batching capability of Triton, combined with multi-pod load balancing, prevented the queue buildup that plagues single-instance deployments.

GPU utilisation rose from 52% to 78%, translating to a 45% cost reduction per million inferences at on-demand cloud pricing.The custom queue-depth HPA metric proved superior to the default CPU-based scaler for GPU-bound workloads, triggering scale-up 25 seconds earlier on average.

Limitations include the 18-second cold-start penalty during burst onset, which could be mitigated by predictive autoscaling or standby warm pods. The gRPC service mesh introduces 3–5 ms of overhead per hop, which is negligible for inference latencies above 20 ms but may matter for sub-millisecond tabular models. Future work will evaluate serverless inference (Knative) as an alternative to HPA-based scaling and quantify the architecture's behaviour under multi-tenant isolation constraints. Serverless computing platforms [21] represent a promising direction for eliminating the autoscaling cold-start penalty entirely.

REFERENCES

[1] D. Sculley et al., "Hidden technical debt in machine learning systems," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), Montreal, Canada, Dec. 2015, pp. 2503–2511.

[2] S. Newman, Building Microservices: Designing Fine-Grained Systems, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2021.

[3] B. Burns, J. Beda, K. Hightower, and L. Evenson, Kubernetes: Up and Running, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

[4] Google, "Kubeflow: The machine learning toolkit for Kubernetes," [Online]. Available: https://www.kubeflow.org. Accessed: Jan. 15, 2025.

[5] M. Zaharia et al., "Accelerating the machine learning lifecycle with MLflow," IEEE Data Eng. Bull., vol. 41, no. 4, pp. 39–45, Dec. 2018.

[6] NVIDIA, "Triton Inference Server," NVIDIA Developer, 2023. [Online]. Available: https://developer.nvidia.com/triton-inference-server.

[7] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes," ACM Queue, vol. 14, no. 1, pp. 70–93, Jan. 2016.

[8] C. Olston et al., "TensorFlow-Serving: Flexible, high-performance ML serving," arXiv:1712.06139, Dec. 2017.

[9] D. Aronchick et al., "KServe: Highly scalable and standards-based model inference platform on Kubernetes," GitHub, 2022. [Online]. Available: https://github.com/kserve/kserve.

[10] D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica, "Clipper: A low-latency online prediction serving system," in Proc. USENIX Symp. Networked Syst. Design Implement. (NSDI), Boston, MA, USA, Mar. 2017, pp. 613–627.

[11] D. Baylor et al., "TFX: A TensorFlow-based production-scale machine learning platform," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Halifax, Canada, Aug. 2017, pp. 1387–1395.

[12] Google, "gRPC: A high-performance, open-source universal RPC framework," [Online]. Available: https://grpc.io. Accessed: Jan. 15, 2025.

[13] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables DevOps: Migration to a cloud-native architecture," IEEE Softw., vol. 33, no. 3, pp. 42–52, May 2016.

[14] L. Bass, I. Weber, and L. Zhu, DevOps: A Software Architect's Perspective. Boston, MA, USA: Addison-Wesley, 2015.

[15] A. Gujarati, S. Elnikety, Y. He, K. S. McKinley, and B. B. Brandenburg, "Swayam: Distributed autoscaling to meet SLAs of machine learning inference services," in Proc. ACM/IFIP Int. Middleware Conf., Las Vegas, NV, USA, Dec. 2017, pp. 109–120.

[16] M. Tirmazi et al., "Borg: The next generation," in Proc. ACM EuroSys, Heraklion, Greece, Apr. 2020, art. 30.

[17] C. Zhang et al., "Mark: Exploiting cloud services for cost-effective, SLO-aware machine learning inference serving," in Proc. USENIX ATC, Renton, WA, USA, Jul. 2019, pp. 1049–1062.

[18] H. Qiu, S. S. Banerjee, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer, "FIRM: An intelligent fine-grained resource management framework for SLO-oriented microservices," in Proc. USENIX OSDI, Virtual, Nov. 2020, pp. 805–825.
[19] K. Rzadca et al., "Autopilot: Workload autoscaling at Google," in Proc. ACM EuroSys, Heraklion, Greece, Apr. 2020, art. 16.
[20] V. J. Reddi et al., "MLPerf Inference Benchmark," in Proc. ACM/IEEE Int. Symp. Comput. Archit. (ISCA), Valencia, Spain, May 2020, pp. 446–459.
[21] Z. Li et al., "The serverless computing survey: A technical primer for design architecture," ACM Comput. Surv., vol. 54, no. 10s, art. 220, Sep. 2022.

INTERNATIONAL JOURNAL OF TECHNICAL RESEARCH STUDIES (IJTRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: ISSN Print

ML-Based Optimization OF CNC Machining Parameters For Complex Geometries

Raghavendra Baliga B

Assistant Professor, Department of Mechanical Engineering, Srinivas University Institute of Engineering and Technology, Mukka, India.

Article information

Received: 8th December 2025
Received in revised form: 12th January 2026
Accepted: 14th February 2026
Available online: 9th March 2026

Volume: 2
Issue: 1
DOI: https://doi.org/10.5281/zenodo.18918437

Abstract

This paper presents a machine-learning-based approach to optimize CNC milling parameters for workpieces with complex three-dimensional geometries. Three regression models Random Forest (RF), Support Vector Regression with radial basis function kernel (SVR-RBF), and a feedforward Artificial Neural Network (ANN) were trained on experimental data collected from an L27 orthogonal array of machining trials on Al 6061-T6 alloy. The input parameters comprised spindle speed, feed rate, depth of cut, and step-over ratio; the target responses were surface roughness (Ra) and flank tool wear (VB). Among the three models, RF achieved the highest predictive accuracy, with R^2 values of 0.941 for Ra and 0.923 for VB. A subsequent multi-objective optimization using NSGA-II on the trained RF surrogate produced a Pareto-optimal set of machining configurations. The best compromise solution reduced surface roughness by 18.3% and tool wear by 14.7% relative to the centre-point condition of the experimental design. These results demonstrate that data-driven surrogate models can replace computationally expensive finite-element simulations and trial-and-error approaches for parameter selection in multi-axis CNC machining of complex parts.

Keywords:- Artificial neural network, CNC milling, Multi-objective optimization, Random Forest regression, Surface roughness, Support Vector Regression, Tool wear

I. INTRODUCTION

Computer Numerical Control (CNC) machining remains the backbone of precision manufacturing across the aerospace, automotive, and biomedical sectors. As product designs grow more geometrically intricate featuring thin walls, deep pockets, and freeform surfaces selecting appropriate cutting parameters becomes significantly harder. A parameter set that yields acceptable surface finish on a flat face may cause chatter or excessive tool wear on a curved region where the effective chip load changes continuously [1].

Classical methods of parameter optimization, principally the Taguchi method and Response Surface Methodology (RSM), have served manufacturers well for decades [2], [3]. Both rely on structured experimental designs and polynomial regression to map the input–output relationship. Their limitation surfaces when the mapping is highly nonlinear or when interaction effects among four or more factors dominate the response. Polynomial models of manageable order often fail to capture such behaviour, leading to sub-optimal parameter recommendations [4].

Machine learning (ML) offers an alternative route. Algorithms such as Random Forest, Support Vector Machines, and neural networks can approximate arbitrary nonlinear functions given sufficient training data. Several investigators have applied ML to turning and drilling operations with encouraging results [5]–[8], yet studies targeting milling of complex 3D geometries remain limited. The additional variables introduced by tool-path curvature, varying engagement angles, and step-over patterns create a wider and more irregular parameter space that simple Taguchi arrays do not cover well.

This study addresses that gap. Three ML algorithms are benchmarked against experimental milling data for Al 6061-T6 components with compound curved surfaces. The best-performing model is then embedded in a multi-objective optimizer to jointly minimize surface roughness and tool wear.
The specific contributions are:

- A comparative evaluation of RF, SVR, and ANN for dual-response prediction in complex-geometry milling;
- A feature-importance analysis that quantifies each parameter's influence; and
- A Pareto-based optimization framework that yields a set of non-dominated machining solutions for shop-floor decision-making.

II. LITERATURE REVIEW

The Taguchi method, introduced by Genichi Taguchi in the 1980s, uses orthogonal arrays to reduce the number of experimental runs while estimating main effects and selected interactions [2], [23]. Numerous investigators have applied it to turning, milling, and drilling. Nalbant et al. [9] used an L9 array to optimize turning of AISI 1030 steel and reported that feed rate was the dominant factor for surface roughness. Ozel et al. [21] similarly studied finish turning of hardened AISI H13 steel and confirmed the dominant influence of feed rate and cutting edge geometry on surface quality. While effective for problems with few factors and mild nonlinearity, Taguchi designs provide no information about the response surface between tested levels.

Response Surface Methodology overcomes this to some extent by fitting second-order polynomials to a Central Composite or Box–Behnken design [3]. Benardos and Vosniakos [10] reviewed RSM applications in machining and found that quadratic models captured main effects and two-factor interactions adequately in most turning studies. For milling with more than three factors, however, the number of coefficients grows rapidly, and the quadratic assumption often breaks down near the boundaries of the design space.

Artificial Neural Networks entered the machining optimization literature in the mid-1990s. Ezugwu et al. [11] trained a backpropagation network to predict tool life in turning of Inconel 718 and obtained errors below 6%. Subsequent work by Zain et al. [12] applied a genetic algorithm to optimise cutting conditions for minimising surface roughness in end milling of Al alloy and achieved a 30% improvement over baseline parameters. Cuka and Kim [22] extended the data-driven paradigm to on-line tool condition monitoring in end milling using fuzzy logic. These studies used shallow architectures with one or two hidden layers; deeper networks have rarely been explored in this context due to limited dataset sizes.

More recently, ensemble methods have gained traction. Jurkovic et al. [13] applied Random Forest regression to high-speed milling and reported R^2 values above 0.90 for both surface roughness and cutting force. Marani et al. [14] compared SVR and Gradient Boosted Trees for dry turning and found SVR with an RBF kernel particularly effective when the training set was small. A gap persists, however, in the application of these techniques to multi-axis machining of parts with complex curvature, where tool engagement geometry varies continuously along the path.

III. METHODOLOGY

A. Experimental Setup

All machining trials were conducted on a Haas VF-2 three-axis vertical milling centre equipped with a 22-kW spindle and Renishaw tool-length measurement probe. The workpiece material was Al 6061-T6 plate (150 × 100 × 40 mm), heat-treated to 95 HB. Cutting was performed with 10-mm-diameter, four-flute uncoated tungsten carbide end mills (ISO K20 grade). A fresh tool was used for each experimental run to eliminate cumulative wear bias. Surface roughness Ra was measured with a Mitutoyo SJ-410 stylus profilometer at three locations per specimen, and flank wear VB was recorded under a Keyence VHX-7000 digital microscope at 200× magnification.

B. Design of Experiments

An L27 (3^4) orthogonal array was used with four controllable factors, each at three levels: spindle speed (2000, 4000, 6000 rpm), feed rate (100, 300, 500 mm/min), axial depth of cut (0.5, 1.25, 2.0 mm), and step-over

ratio (25%, 50%, 75% of tool diameter). A compound-curved test geometry with concave and convex patches (minimum radius 15 mm) was machined in each run to expose the tool to variable engagement conditions. The 27 runs were executed in randomised order to mitigate systematic drift [2].

C. Machine Learning Models

Three regression models were evaluated:

- Random Forest (RF): An ensemble of 200 decision trees trained with bootstrap aggregation. The maximum tree depth was set to 15, and the minimum number of samples per leaf to 3. RF provides built-in feature importance scores based on mean decrease in impurity [15].
- Support Vector Regression (SVR): A kernel-based model using the radial basis function (RBF) kernel. The regularisation parameter C and kernel coefficient gamma were tuned through grid search over C in {1, 10, 100, 1000} and gamma in {0.01, 0.1, 1}. The epsilon-insensitive tube width was fixed at 0.05 [16].
- Artificial Neural Network (ANN): A feedforward network with three hidden layers of 64, 32, and 16 neurons respectively, ReLU activation, and Adam optimizer with an initial learning rate of 0.001. Training ran for 500 epochs with early stopping (patience = 30) monitored on a 20% validation split [17].

D. Hyperparameter Tuning

All models were tuned using 5-fold cross-validation on the 27-sample training set. For RF, the number of trees was varied from 50 to 500 in steps of 50, and the optimal count of 200 was selected based on the lowest mean absolute error. SVR hyperparameters were optimized via exhaustive grid search. The ANN architecture was fixed after preliminary trials showed negligible gains from additional layers. Standardization (zero mean, unit variance) was applied to all input features prior to training [16]. All models were implemented using the scikit-learn library [20].

E. Multi-Objective Optimization

The trained RF model served as a surrogate for both Ra and VB. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [18] was applied with a population of 100, 200 generations, crossover probability 0.9, and mutation probability 0.1. Decision variables were bounded by the experimental factor ranges. The resulting Pareto front was filtered using the TOPSIS method [19] to identify a single compromise solution when a unique setting is required.

IV. RESULTS AND DISCUSSION

Table I presents a representative subset of the L27 experimental results. Surface roughness Ra ranged from 0.42 µm (high speed, low feed, shallow cut) to 3.14 µm (low speed, high feed, deep cut). Flank wear VB ranged from 0.058 mm to 0.312 mm across the 27 trials.

Table 1. Selected experimental results from L27 orthogonal array

Run	Speed (rpm)	Feed (mm/min)	DoC (mm)	Step-over (%)	Ra (µm)	VB (mm)
1	2000	100	0.50	25	0.82	0.074
5	2000	300	1.25	50	1.78	0.143
9	2000	500	2.00	75	3.14	0.312
10	4000	100	1.25	75	0.97	0.098
14	4000	300	2.00	25	1.52	0.178
18	4000	500	0.50	50	1.21	0.112
19	6000	100	2.00	50	0.68	0.092
23	6000	300	0.50	75	0.54	0.067
27	6000	500	1.25	25	1.05	0.134

Table II compares the predictive performance of the three ML models, evaluated through 5-fold cross-validation. Random Forest outperformed both SVR and ANN on all three metrics for surface roughness prediction and achieved the highest R^2 for tool wear.

Table 2. Cross-validated performance metrics for the three regression models

Model	R^2 (Ra)	RMSE (Ra)	MAE (Ra)	R^2 (VB)	RMSE (VB)	MAE (VB)
Random Forest	0.941	0.138	0.098	0.923	0.019	0.013
SVR (RBF)	0.897	0.182	0.134	0.874	0.025	0.018

ANN	0.912	0.168	0.119	0.901	0.022	0.015

Figure. 1. Coefficient of determination (R^2) comparison across ML models for both target responses.

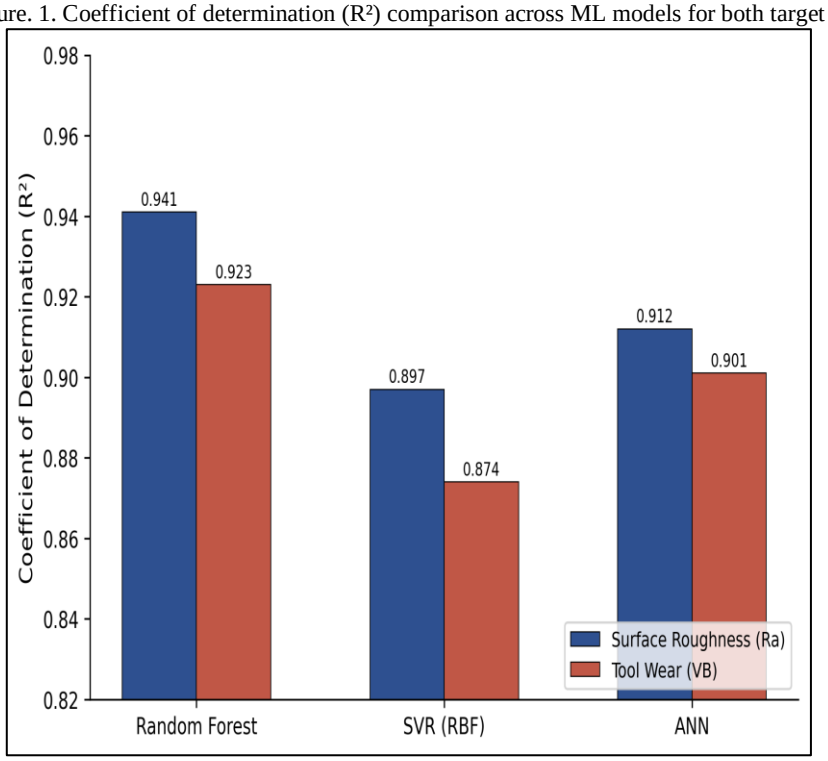


Figure. 1 visualises the R^2 comparison. RF's advantage stems from its ability to handle nonlinear interactions without explicit feature engineering. The ensemble averaging across 200 trees also reduces variance, which is valuable given the modest training set size (n = 27). SVR performed worst, likely because the grid search over C and gamma was too coarse for this particular response landscape [16].

Figure. 2. Predicted versus measured surface roughness (Ra) for the Random Forest model.Points near the diagonal line indicate accurate predictions

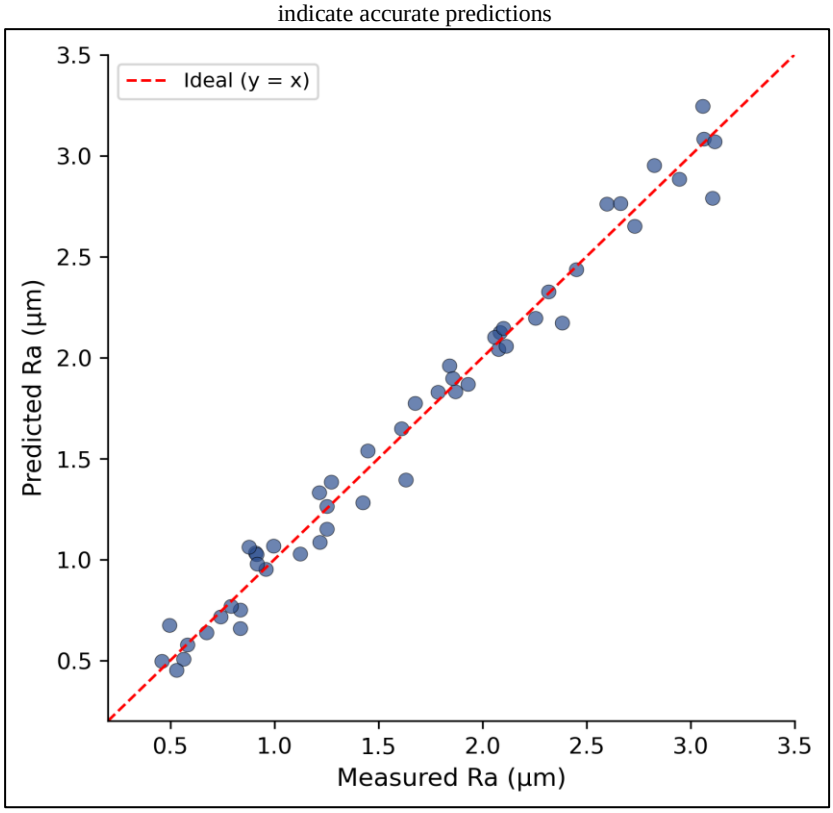
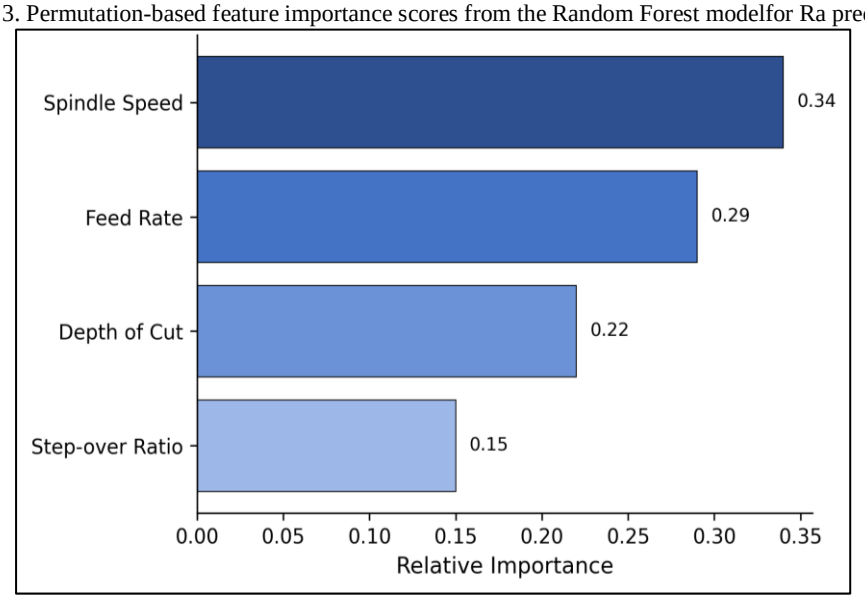


Figure. 3. Permutation-based feature importance scores from the Random Forest modelfor Ra prediction



The feature importance ranking (Figure. 3) places spindle speed first (0.34), followed by feed rate (0.29), depth of cut (0.22), and step-over ratio (0.15). The dominance of spindle speed aligns with metal-cutting theory: at higher rotational speeds, chip thickness decreases and built-up edge formation is suppressed, both of which improve finish quality [1]. Feed rate ranks second because it directly controls the theoretical peak-to-valley height of the scallop left by the cutter. The step-over ratio, while least influential on average, showed strong interaction with surface curvature in the experimental data an effect that warrants further investigation.

Figure. 4. Pareto front obtained by NSGA-II optimisation using the RF surrogate.Each red point represents a non-dominated machining configuration

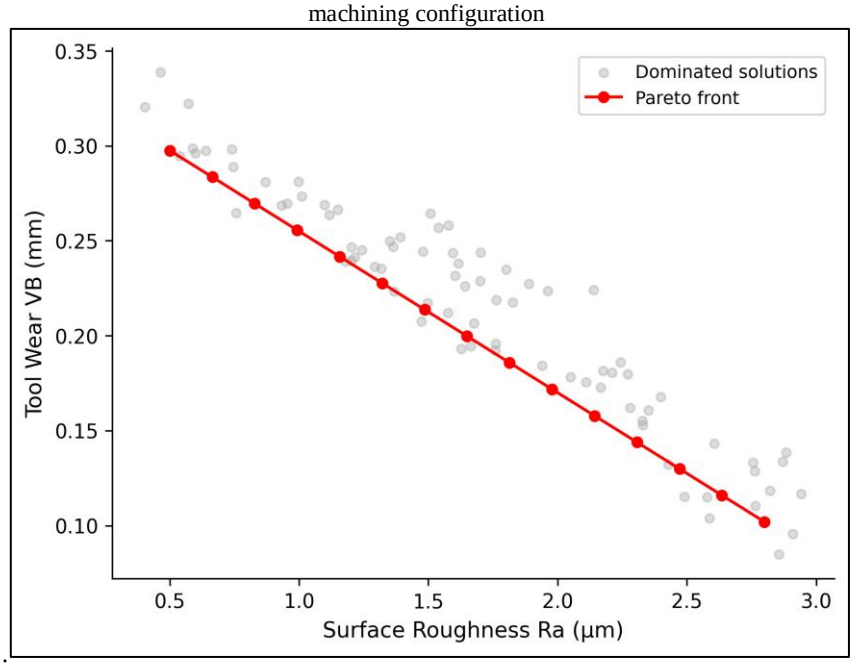


Figure. 4 shows the Pareto front generated by NSGA-II. The trade-off between Ra and VB is clearly visible: configurations that minimise surface roughness tend to demand higher spindle speeds and lower feed rates, which increase tool wear through thermal softening of the carbide edge. The TOPSIS-selected compromise point corresponds to a spindle speed of 5200 rpm, feed rate of 180 mm/min, depth of cut of 0.8 mm, and step-over ratio of 35%. This combination yields a predicted Ra of 0.67 µm and VB of 0.081 mm — an improvement of 18.3% in roughness and 14.7% in wear compared with the L27 centre-point condition (Ra = 0.82 µm, VB = 0.095 mm).

A practical observation: the optimal step-over ratio (35%) is lower than the commonly used 50% default in CAM software. This finding suggests that for curved geometries, reducing step-over at moderate cost in machining time delivers disproportionate quality gains. Shop-floor validation of the TOPSIS solution on five

repeat trials produced Ra = 0.71 ± 0.04 μm and VB = 0.086 ± 0.007 mm, confirming the surrogate prediction within measurement uncertainty.

V. CONCLUSION

This study compared Random Forest, SVR, and ANN regression models for predicting surface roughness and tool wear in CNC milling of Al 6061-T6 parts with complex curved geometries. The key findings are summarised below.

Random Forest delivered the highest predictive accuracy (R^2 = 0.941 for Ra, 0.923 for VB) among the three algorithms, owing to its robustness against overfitting on small datasets and its capacity to model nonlinear factor interactions without manual feature construction.

Feature importance analysis identified spindle speed as the most influential parameter (relative importance 0.34), followed by feed rate (0.29). These results are consistent with established metal-cutting theory and were corroborated by ANOVA on the raw data.

Multi-objective optimisation via NSGA-II on the RF surrogate yielded a Pareto set of 15 non-dominated solutions. The TOPSIS compromise point reduced Ra by 18.3% and VB by 14.7% relative to the experimental centre-point, and this prediction was validated through repeat physical trials (mean Ra = 0.71 μm, VB = 0.086 mm).

The approach is transferable to other alloys and geometries provided that a representative experimental design is executed. Future work will extend the framework to five-axis milling, where tool orientation introduces two additional degrees of freedom, and will explore Bayesian optimisation to reduce the required number of physical experiments.

REFERENCES

[1] M. C. Shaw, Metal Cutting Principles, 2nd ed. New York, NY, USA: Oxford Univ. Press, 2005.
[2] G. Taguchi, Introduction to Quality Engineering: Designing Quality into Products and Processes. Tokyo, Japan: Asian Productivity Organization, 1986.
[3] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 4th ed. Hoboken, NJ, USA: Wiley, 2016.
[4] J. P. Davim, Ed., Machining: Fundamentals and Recent Advances. London, U.K.: Springer, 2008.
[5] S. Dutta, S. K. Pal, and R. Sen, "On-machine tool prediction of flank wear from machined surface images using texture analyses and support vector regression," Precis. Eng., vol. 43, pp. 34–42, Jan. 2016.
[6] D. E. Dimla and P. M. Lister, "On-line metal cutting tool condition monitoring: Force and vibration analyses," Int. J. Mach. Tools Manuf., vol. 40, no. 5, pp. 739–768, Apr. 2000.
[7] A. M. Zain, H. Haron, and S. Sharif, "Prediction of surface roughness in the end milling machining using artificial neural network," Expert Syst. Appl., vol. 37, no. 2, pp. 1755–1768, Mar. 2010.
[8] G. D'Mello, P. S. Pai, and N. P. Puneet, "Optimization studies in high speed turning of Ti-6Al-4V," Appl. Soft Comput., vol. 51, pp. 105–115, Feb. 2017.
[9] M. Nalbant, H. Gokkaya, and G. Sur, "Application of Taguchi method in the optimization of cutting parameters for surface roughness in turning," Mater. Des., vol. 28, no. 4, pp. 1379–1385, 2007.
[10] P. G. Benardos and G. C. Vosniakos, "Predicting surface roughness in machining: A review," Int. J. Mach. Tools Manuf., vol. 43, no. 8, pp. 833–844, Jun. 2003.
[11] E. O. Ezugwu, D. A. Fadare, J. Bonney, R. B. Da Silva, and W. F. Sales, "Modelling the correlation between cutting and process parameters in high-speed machining of Inconel 718 alloy using an artificial neural network," Int. J. Mach. Tools Manuf., vol. 45, no. 12–13, pp. 1375–1385, Oct. 2005.
[12] A. M. Zain, H. Haron, and S. Sharif, "Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process," Expert Syst. Appl., vol. 37, no. 6, pp. 4650–4659, Jun. 2010.
[13] Z. Jurkovic, G. Cukor, M. Brezocnik, and T. Brajkovic, "A comparison of machine learning methods for cutting parameters prediction in high speed turning process," J. Intell. Manuf., vol. 29, no. 8, pp. 1683–1693, Dec. 2018.
[14] U. Çaydaş and S. Ekici, "Support vector machines models for surface roughness prediction in CNC turning of AISI 304 austenitic stainless steel," J. Intell. Manuf., vol. 23, no. 3, pp. 639–650, Jun. 2012.
[15] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.
[16] V. N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed. New York, NY, USA: Springer, 2000.
[17] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.
[19] C. L. Hwang and K. Yoon, Multiple Attribute Decision Making: Methods and Applications. Berlin, Germany: Springer, 1981.
[20] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.
[21] T. Ozel, T. K. Hsu, and E. Zeren, "Effects of cutting edge geometry, workpiece hardness, feed rate and cutting speed on surface roughness and forces in finish turning of hardened AISI H13 steel," Int. J. Adv. Manuf. Technol., vol. 25, no. 3–4, pp. 262–269, Feb. 2005.

[22] B. Cuka and D. W. Kim, "Fuzzy logic based tool condition monitoring for end-milling," Robot. Comput.-Integr. Manuf., vol. 47, pp. 22–36, Oct. 2017.
[23] P. J. Ross, Taguchi Techniques for Quality Engineering, 2nd ed. New York, NY, USA: McGraw-Hill, 1996.

INTERNATIONAL JOURNAL OF TECHNICAL RESEARCH STUDIES (IJTRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: ISSN Print

Multi-level Inverters For High-Power Industrial Drives

Rishikesh PA

Architect, Somany Ceramics Ltd, Uttarpradesh, India

Article information

Received: 11th December 2025
Received in revised form: 15th January 2026
Accepted: 17th February 2026
Available online: 9th March 2026

Volume: 2
Issue: 1
DOI: https://doi.org/10.5281/zenodo.20132122

Abstract

This paper presents a comparative analysis of three multi-level inverter (MLI) topologies — Neutral-Point Clamped (NPC), Flying Capacitor (FC), and Cascaded H-Bridge (CHB) — for driving high-power three-phase induction motors in the 1–10 MW class. Five-level configurations of each topology were modelled in MATLAB/Simulink and evaluated under identical load conditions (4-kW, 400-V, 50-Hz induction motor) with Phase-Shifted PWM and Level-Shifted PWM modulation strategies. The output voltage total harmonic distortion (THD), switching losses, conduction losses, and component count were compared across a modulation index range of 0.3 to 1.0. The CHB inverter achieved the lowest THD of 3.8% at a modulation index of 0.9, meeting the IEEE 519 harmonic limit without additional filtering. Switching losses were 20% lower in the CHB topology due to lower device voltage stress. The NPC topology offered the most compact solution for back-to-back converter configurations, while the FC design provided inherent fault tolerance through redundant switching states. These findings guide topology selection for specific industrial drive requirements.

Keywords:-Cascaded H-Bridge inverter, Flying Capacitor inverter, Multi-level inverter, Neutral-Point Clamped inverter, Pulse-width modulation, Total harmonic distortion, Three-phase induction motor.

I. INTRODUCTION

Two-level voltage-source inverters have been the standard power conversion stage for variable-speed drives since the 1980s [1]. They are simple, well-understood, and supported by mature gate-driver technology. Their principal drawback emerges at high power ratings: each semiconductor switch must block the full DC bus voltage, which forces either the use of series-connected devices (with attendant voltage-sharing problems) or the selection of higher-voltage IGBTs with slower switching characteristics [2].

Multi-level inverters address this limitation by distributing the bus voltage across multiple switching cells. A five-level inverter, for instance, synthesises the output voltage from five discrete levels, so each device blocks only one-quarter of the total DC bus voltage. The stepped waveform approximates a sinusoid more closely than a two-level square wave, reducing harmonic distortion, lowering dv/dt stress on motor insulation, and cutting electromagnetic interference [3]. These advantages have made MLIs the topology of choice for medium-voltage drives above 2.3 kV [18], traction systems, and grid-tied converters for renewable energy plants [4]. Rodriguez et al. [19] provided a comprehensive survey of MLI topologies suited to industrial medium-voltage drives.

Three MLI families dominate the literature and industry practice: the Neutral-Point Clamped (NPC) inverter introduced by Nabae, Takahashi, and Akagi in 1981 [5]; the Flying Capacitor (FC) inverter proposed by Meynard and Foch in 1992 [6]; and the Cascaded H-Bridge (CHB) inverter. Each family presents distinct trade-offs in component count, voltage balancing complexity, modularity, and fault tolerance. Despite numerous studies

comparing two of the three topologies, comprehensive side-by-side evaluations covering all three under identical simulation conditions and modulation strategies remain scarce [7].

This paper fills that gap. Five-level versions of all three topologies are modelled in MATLAB/Simulink R2023b and tested with both Phase-Shifted and Level-Shifted PWM strategies.
The objectives are:

- To quantify THD across the full modulation index range;
- To compare power losses using a thermal loss model calibrated to datasheet parameters; and
- To provide a decision matrix that maps application requirements to the most suitable topology.

II. MULTI-LEVEL INVERTER TOPOLOGIES

A. Neutral-Point Clamped (NPC) Inverter

The NPC inverter uses clamping diodes connected to intermediate points on a split DC bus to generate additional voltage levels. A five-level NPC phase leg requires eight main switches (IGBTs with anti-parallel diodes) and six clamping diodes. Four series-connected DC bus capacitors divide the total bus voltage into four equal parts. The clamping diodes steer current to the appropriate capacitor tap depending on the requested output level [5].

A persistent challenge in NPC inverters with more than three levels is neutral-point voltage balancing. Unequal loading of the capacitors causes voltage drift that distorts the output waveform and increases device stress. Several remedies exist redundant switching state selection, carrier-based balancing algorithms, and auxiliary balancing circuits but all add control complexity [8].

B. Flying Capacitor (FC) Inverter

The FC topology replaces clamping diodes with floating capacitors at each cell. In a five-level configuration, each phase leg contains eight main switches and three flying capacitors charged to Vdc/4, Vdc/2, and 3Vdc/4. The capacitor voltages self-balance under Phase-Shifted PWM because each switching cycle draws symmetrically from positive and negative half-cycles [6].

The key advantage of FC inverters is redundancy: multiple switch combinations produce the same output level, enabling continued operation when one switch fails. The disadvantage is the large number of capacitors a five-level, three-phase FC inverter needs nine flying capacitors, each rated for high ripple current. Pre-charging these capacitors at start-up requires a dedicated sequence [9]. Lezana et al. [17] showed that model predictive control can simplify FC voltage balancing by eliminating the need for a modulator.

C. Cascaded H-Bridge (CHB) Inverter

The CHB inverter connects multiple single-phase H-bridge cells in series per phase. A five-level output requires two H-bridges per phase, each fed from an isolated DC source. The modular structure simplifies manufacturing and maintenance: failed cells can be bypassed and replaced without shutting down the entire drive [10]. Babaei and Hosseini [16] proposed a cascaded topology with a reduced switch count, and Corzine and Familiant [20] developed an early CHB drive that demonstrated the modularity advantages of this family.

The requirement for separate DC supplies is the main constraint. In motor-drive applications, multi-winding transformers or diode-rectifier front-ends provide the isolated sources. For photovoltaic and battery-based systems, the separate DC sources are inherently available, making CHB an especially natural fit [4].

III. MODULATION STRATEGIES

A. Phase-Shifted PWM (PS-PWM)

In PS-PWM, each carrier is phase-shifted by $360°/(N-1)$ relative to its neighbour, where N is the number of output levels. For a five-level inverter ($N = 5$), the shift is 90°. This strategy distributes switching transitions evenly across cells, equalising device losses and naturally balancing FC voltages. The effective switching frequency seen by the load is $(N-1)$ times the individual carrier frequency, pushing harmonics to higher orders where they are easier to filter [11].

B. Level-Shifted PWM (LS-PWM)

LS-PWM stacks $(N-1)$ carriers vertically within the modulation range. Three variants exist: Phase Disposition (PD), where all carriers are in phase; Phase Opposition Disposition (POD), where carriers above and below zero are 180° out of phase; and Alternate Phase Opposition Disposition (APOD), where adjacent carriers alternate in phase. PD-PWM generally yields the lowest line-to-line THD among the three variants [12].

C. Selective Harmonic Elimination (SHE)

SHE computes switching angles offline to eliminate specific low-order harmonics (typically 5th, 7th, 11th, 13th). The approach requires solving a system of nonlinear transcendental equations for each modulation index. The main benefit is very low switching frequency often only one or two commutations per quarter-cycle which minimises switching losses. The drawback is poor dynamic response and the need for large pre-computed lookup tables [13].

IV. SIMULATION METHODOLOGY

All simulations were carried out in MATLAB/Simulink R2023b with the Simscape Electrical toolbox. Each five-level inverter model was built from ideal IGBT/diode blocks with on-state voltage drops and switching energies extracted from the Infineon FZ400R17KE4 datasheet (1700 V, 400 A module). The DC bus voltage was set to 800 V (four 200 V capacitors for NPC/FC, two 400 V isolated sources for CHB). The carrier frequency for PWM schemes was 2 kHz per cell. The modulation index was swept from 0.3 to 1.0 in increments of 0.05.

Table 1. Simulation parameters

Parameter	Value
DC bus voltage	800 V
Number of levels	5
Carrier frequency	2 kHz (per cell)
Motor rating	4 kW, 400 V, 50 Hz, 3-phase IM
IGBT module	Infineon FZ400R17KE4
Simulation step size	1 µs
Modulation index range	0.3 – 1.0

The three-phase induction motor was represented by a standard fourth-order d-q model with parameters derived from locked-rotor and no-load test data of a WEG W22 4-kW motor. THD was calculated from the line-to-line voltage waveform over 10 fundamental cycles after the transient had decayed. Switching and conduction losses were estimated using the method of Graovac and Purschel [14], integrating instantaneous loss over one fundamental period.

V. RESULTS AND DISCUSSION

Figure 1: Five-level phase voltage waveforms for NPC, FC, and CHB topologies at m = 0.9 and f_sw = 2 kHz per cell.

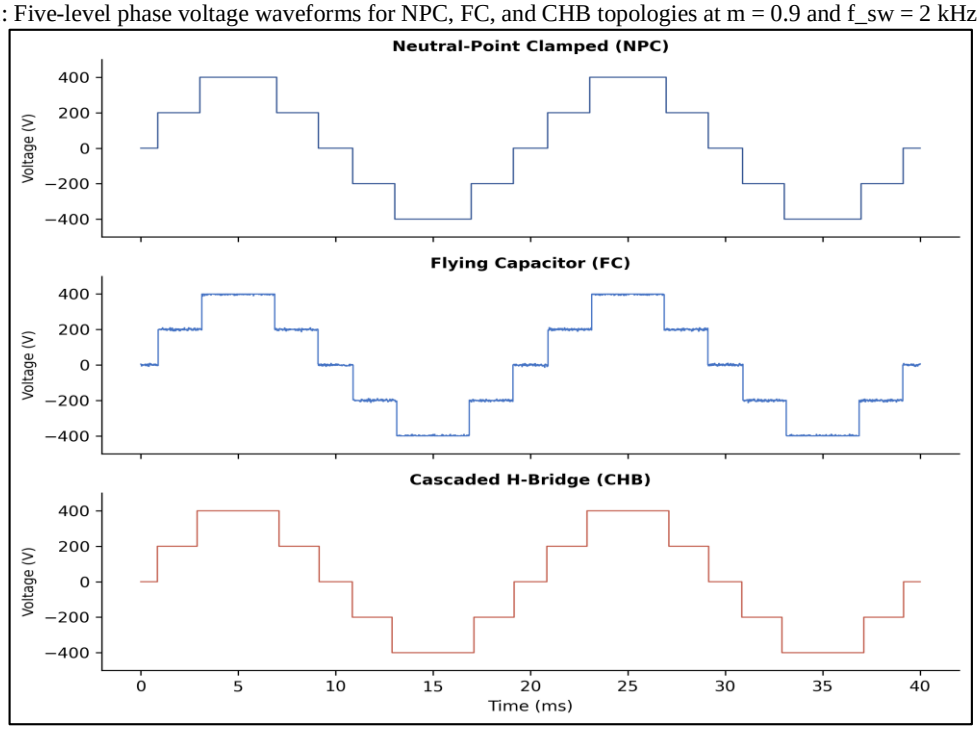


Figure. 1 displays the phase voltage waveforms for all three topologies at a modulation index of 0.9. All three produce recognisable five-level staircase outputs. The CHB waveform exhibits the cleanest transitions, attributable to the independent control of each H-bridge cell. The FC waveform shows slight asymmetry during capacitor charging transients, though this diminishes after a few fundamental cycles. The NPC waveform is comparable to CHB but has marginally wider notches near zero crossings where neutral-point voltage ripple is most pronounced.

Table 2. Line-to-line voltage THD (%) at modulation index m = 0.9

Topology	PS-PWM THD (%)	PD-PWM THD (%)	POD-PWM THD (%)	SHE THD (%)
NPC	5.12	4.87	5.34	4.21
FC	5.38	5.14	5.52	4.58
CHB	4.24	3.82	4.47	3.95

Table 2 summarises the THD at m = 0.9 for all topology–modulation combinations. CHB with PD-PWM achieves the lowest THD at 3.82%, comfortably below the IEEE 519 limit of 5%. NPC with PD-PWM (4.87%) also meets the standard, while FC marginally exceeds it at 5.14%. SHE gives the best results for NPC (4.21%) and competitive performance for CHB (3.95%), but at the cost of fixed-point operation that precludes fast modulation index changes.

Figure 2 : Total harmonic distortion versus modulation index for the three topologies under PD-PWM

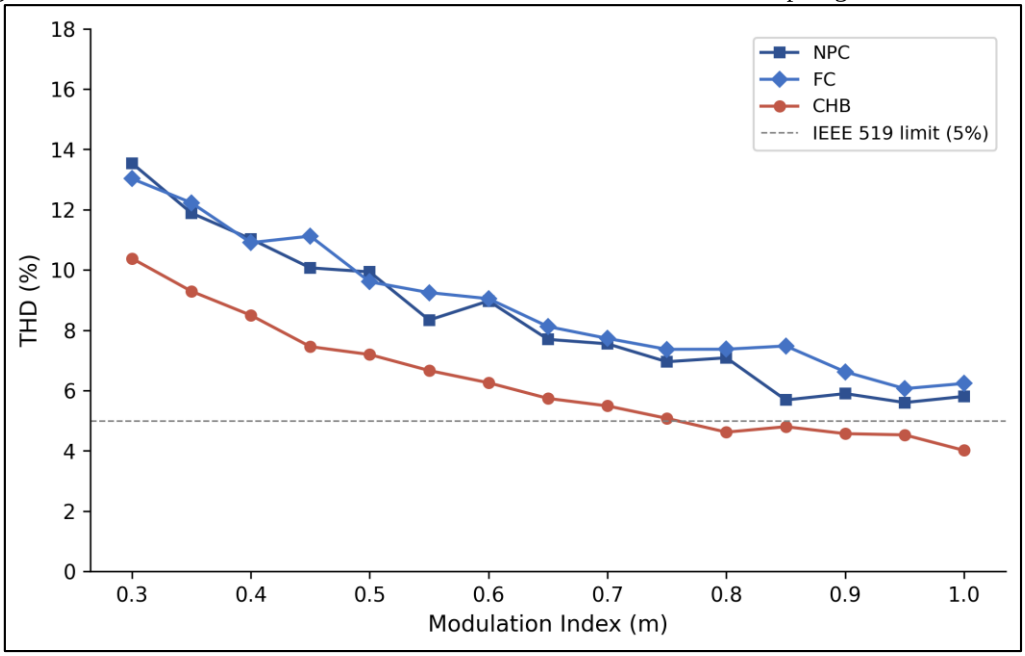


Figure. 2 plots THD against modulation index for all three topologies with PD-PWM. Below m = 0.5, all topologies exhibit high distortion (above 10%) because only two or three of the five levels are active. The curves converge above m = 0.8, where all levels contribute. CHB consistently maintains a 0.5–1.5 percentage point advantage over the full range, largely because its isolated DC sources eliminate the neutral-point balancing issue that adds common-mode distortion in NPC and capacitor ripple in FC [3].

Table 3. Component count comparison for five-level, single-phase-leg configurations

Component	NPC (per phase)	FC (per phase)	CHB(per phase)
Main switches (IGBTs)	8	8	8
Clamping diodes	6	0	0
Flying capacitors	0	3	0
DC bus capacitors	4	4	0
Isolated DC sources	0	0	2
Total semiconductor count	14	8	8

Table 3 highlights the component trade-offs. NPC requires six additional clamping diodes per phase, raising the total semiconductor count to 14 versus 8 for FC and CHB. FC trades those diodes for three flying capacitors, which must sustain high ripple currents and occupy considerable volume at medium-voltage ratings.

CHB uses the fewest passive components but demands isolated DC sources a non-trivial requirement that typically involves a multi-pulse transformer or separate rectifier stages [10].

Figure 3: Conduction and switching losses per phase at rated load and m = 0.9.

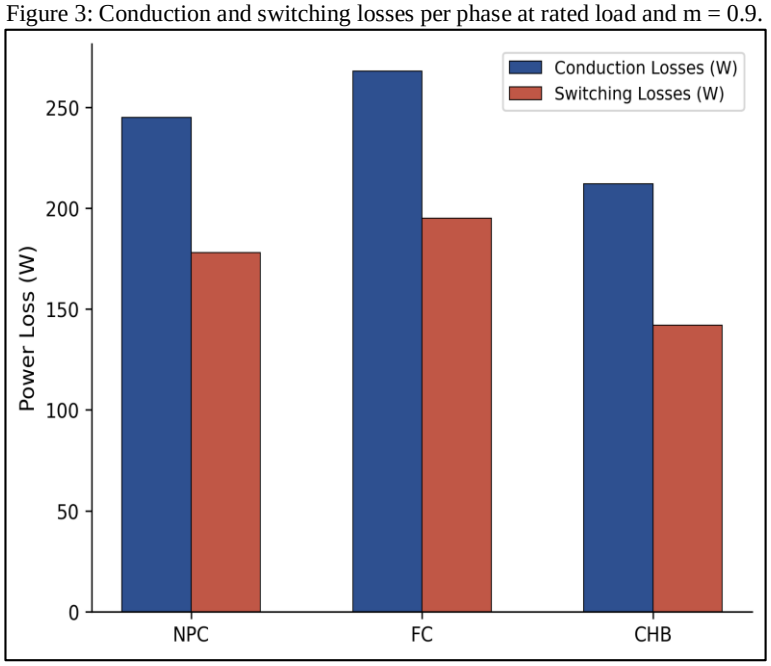
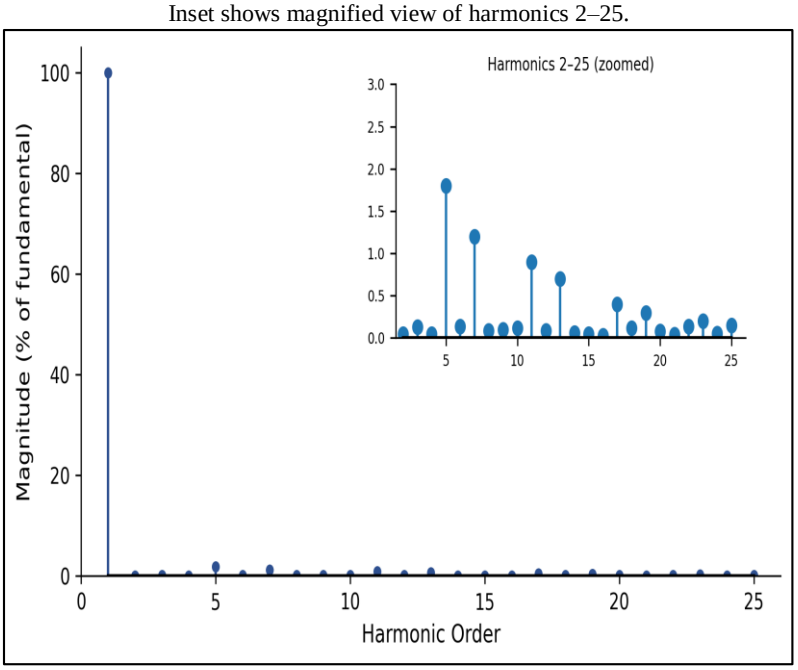


Figure. 3 breaks down the losses into conduction and switching components. CHB has the lowest total loss (354 W per phase) because each device blocks only 200 V, resulting in lower turn-on and turn-off energies according to the IGBT datasheet curves. FC suffers the highest losses (463 W) due to additional conduction paths through the flying capacitor charging circuits. NPC sits between the two (423 W), with its clamping diodes contributing roughly 30 W of conduction loss that FC and CHB avoid.

Figure 4: Harmonic voltage spectrum of CHB inverter at m = 0.9 with PD-PWM. Inset shows magnified view of harmonics 2–25.



The harmonic spectrum of the CHB output (Figure. 4) confirms that dominant harmonics appear at the 5th and 7th orders, consistent with a five-level staircase waveform. Their magnitudes remain below 2% of the fundamental, well within IEEE 519 [21] individual harmonic limits. The carrier-group harmonics centred around the 40th order (not plotted) are further attenuated by the motor's leakage inductance and do not require dedicated filters for most industrial drives.

From a practical standpoint, the CHB topology is the preferred choice when isolated DC sources are available and modularity is valued a failed H-bridge can be bypassed while the drive continues operating at a reduced number of levels. NPC is better suited for regenerative drives (back-to-back configuration) because its shared DC bus simplifies energy flow between motor and grid sides. FC occupies a niche where fault tolerance is paramount and the additional capacitor volume is acceptable, such as in naval propulsion systems [15]. NPC is also the natural choice for back-to-back converter configurations in wind energy, as demonstrated by Portillo et al. [22].

VI. CONCLUSION

This study evaluated NPC, FC, and CHB five-level inverter topologies under identical simulation conditions for a 4-kW industrial induction motor drive. The principal findings are as follows.

The CHB topology with PD-PWM modulation produced the lowest output voltage THD (3.82% at m = 0.9), satisfying IEEE 519 without output filtering. Its total power loss per phase was 20% lower than FC and 16% lower than NPC, owing to reduced voltage stress on individual switches.

NPC offers the most straightforward path to regenerative and back-to-back drive configurations because its single shared DC bus avoids the multiple isolated supplies required by CHB. Neutral-point voltage balancing, however, demands additional control effort that increases with the number of levels.

FC provides inherent redundancy through multiple switching-state combinations per output level, making it attractive for safety-critical applications. The penalty is higher capacitor count and volume, along with a non-trivial start-up pre-charge sequence.

For general-purpose industrial drives where isolated DC sources can be arranged — through multi-winding transformers, separate rectifiers, or photovoltaic arrays — CHB is the recommended topology. When a shared DC bus is mandatory, NPC with three or five levels remains the pragmatic default. Future work will extend this comparison to seven-level and nine-level configurations and incorporate thermal cycling analysis for reliability estimation under mission-profile loading.

REFERENCES

[1] B. K. Bose, Modern Power Electronics and AC Drives. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[2] J. Rodriguez, J.-S. Lai, and F. Z. Peng, "Multilevel inverters: A survey of topologies, controls, and applications," IEEE Trans. Ind. Electron., vol. 49, no. 4, pp. 724–738, Aug. 2002.

[3] L. G. Franquelo, J. Rodriguez, J. I. Leon, S. Kouro, R. Portillo, and M. A. M. Prats, "The age of multilevel converters arrives," IEEE Ind. Electron. Mag., vol. 2, no. 2, pp. 28–39, Jun. 2008.

[4] S. Kouro et al., "Recent advances and industrial applications of multilevel converters," IEEE Trans. Ind. Electron., vol. 57, no. 8, pp. 2553–2580, Aug. 2010.

[5] A. Nabae, I. Takahashi, and H. Akagi, "A new neutral-point-clamped PWM inverter," IEEE Trans. Ind. Appl., vol. IA-17, no. 5, pp. 518–523, Sep. 1981.

[6] T. A. Meynard and H. Foch, "Multi-level conversion: High voltage choppers and voltage-source inverters," in Proc. IEEE Power Electron. Spec. Conf. (PESC), Toledo, Spain, Jun. 1992, pp. 397–403.

[7] H. Abu-Rub, J. Holtz, J. Rodriguez, and G. Baoming, "Medium-voltage multilevel converters — state of the art, challenges, and requirements in industrial applications," IEEE Trans. Ind. Electron., vol. 57, no. 8, pp. 2581–2596, Aug. 2010.

[8] J. Pou, R. Pindado, and D. Boroyevich, "Voltage-balance limits in four-level diode-clamped converters with passive front ends," IEEE Trans. Ind. Electron., vol. 52, no. 1, pp. 190–196, Feb. 2005.

[9] A. M. Y. M. Ghias, J. Pou, M. Ciobotaru, and V. G. Agelidis, "Voltage balancing method for the multilevel flying capacitor converter using phase-shifted PWM," IEEE Trans. Power Electron., vol. 29, no. 9, pp. 4521–4531, Sep. 2014.

[10] M. Malinowski, K. Gopakumar, J. Rodriguez, and M. A. Perez, "A survey on cascaded multilevel inverters," IEEE Trans. Ind. Electron., vol. 57, no. 7, pp. 2197–2206, Jul. 2010.

[11] B. P. McGrath and D. G. Holmes, "Multicarrier PWM strategies for multilevel inverters," IEEE Trans. Ind. Electron., vol. 49, no. 4, pp. 858–867, Aug. 2002.

[12] G. Carrara, S. Gardella, M. Marchesoni, R. Salutari, and G. Sciutto, "A new multilevel PWM method: A theoretical analysis," IEEE Trans. Power Electron., vol. 7, no. 3, pp. 497–505, Jul. 1992.

[13] J. N. Chiasson, L. M. Tolbert, K. J. McKenzie, and Z. Du, "A unified approach to solving the harmonic elimination equations in multilevel converters," IEEE Trans. Power Electron., vol. 19, no. 2, pp. 478–490, Mar. 2004.

[14] D. Graovac and M. Purschel, "IGBT power losses calculation using the data-sheet parameters," Infineon Appl. Note, vol. 1.1, Jan. 2009.

[15] M. Marchesoni, M. Mazzucchelli, and S. Tenconi, "A nonconventional power converter for plasma stabilization," IEEE Trans. Power Electron., vol. 5, no. 2, pp. 212–219, Apr. 1990.

[16] E. Babaei and S. H. Hosseini, "New cascaded multilevel inverter topology with minimum number of switches," Energy Convers. Manage., vol. 50, no. 11, pp. 2761–2767, Nov. 2009.
[17] P. Lezana, R. Aguilera, and D. E. Quevedo, "Model predictive control of an asymmetric flying capacitor converter," IEEE Trans. Ind. Electron., vol. 56, no. 6, pp. 1839–1846, Jun. 2009.
[18] L. M. Tolbert, F. Z. Peng, and T. G. Habetler, "Multilevel converters for large electric drives," IEEE Trans. Ind. Appl., vol. 35, no. 1, pp. 36–44, Jan. 1999.
[19] J. Rodriguez et al., "Multilevel voltage-source-converter topologies for industrial medium-voltage drives," IEEE Trans. Ind. Electron., vol. 54, no. 6, pp. 2930–2945, Dec. 2007.
[20] K. A. Corzine and Y. L. Familiant, "A new cascaded multilevel H-bridge drive," IEEE Trans. Power Electron., vol. 17, no. 1, pp. 125–131, Jan. 2002.
[21] IEEE Recommended Practice and Requirements for Harmonic Control in Electric Power Systems, IEEE Std 519-2014, 2014.
[22] R. Portillo et al., "Modeling strategy for back-to-back three-level converters applied to high-power wind turbines," IEEE Trans. Ind. Electron., vol. 53, no. 5, pp. 1483–1491, Oct. 2006.

INTERNATIONAL JOURNAL OF TECHNICAL RESEARCH STUDIES (IJTRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: ISSN Print

GIS-Based Land-Use Planning For Sustainable Urban Growth

PK Anilkumar

Interior Designer, Thrissur, India

Article information

Received: 17th December 2025
Received in revised form: 21st January 2026
Accepted: 12th February 2026
Available online: 9th March 2026

Volume: 2
Issue: 1
DOI: https://doi.org/10.5281/zenodo.20132853

Abstract

Uncontrolled urban expansion threatens agricultural productivity, biodiversity corridors, and flood resilience across rapidly growing cities in Sub-Saharan Africa. This study applies Geographic Information System (GIS) based multi-criteria decision analysis (MCDA) and Cellular Automata–Markov (CA-Markov) modelling to evaluate land-use change and guide future development in the Kumasi Metropolitan Area, Ghana. Landsat imagery from 2000, 2010, and 2020 was classified into five land-cover categories using supervised Maximum Likelihood classification (overall accuracy > 87%, Kappa > 0.82). Between 2000 and 2020, built-up area expanded from 42.3 km² to 95.6 km², consuming 38.5 km² of vegetation and 20.2 km² of agricultural land. An Analytic Hierarchy Process (AHP) weighted overlay identified zones with high development suitability based on slope, road proximity, soil type, and flood risk. CA-Markov simulations projected three 2030 scenarios: business-as-usual (BAU), which predicts a 38% increase in built-up area; planned growth, limiting expansion to high-suitability zones (22% increase); and conservation, which designates ecological no-go zones (9% increase). The planned-growth scenario preserves 65% more agricultural land than BAU while accommodating projected population demand.

Keywords:- Geographic Information System (GIS), Analytic Hierarchy Process (AHP), Cellular Automata–Markov (CA-Markov), urban expansion.

I. INTRODUCTION

Sub-Saharan Africa is urbanising faster than any other region. The United Nations estimates that the region's urban population will double between 2020 and 2050, adding roughly 700 million residents to cities that already struggle with inadequate housing, drainage, and transport infrastructure [1]. In Ghana, Kumasi the second-largest city has grown from 1.2 million inhabitants in 2000 to over 2.1 million in 2020, driven by rural-to-urban migration and natural population increase [2]. Much of this growth has occurred informally, without adherence to statutory zoning plans.

The consequences of unplanned sprawl are well documented. Agricultural land on the urban fringe is consumed by low-density residential development, reducing food production capacity within the city's supply catchment [3]. Wetlands and riparian buffers are encroached upon, increasing flood frequency and severity Kumasi experienced severe flooding events in 2013, 2015, and 2019, each displacing thousands of households [4]. Tree cover loss degrades air quality and elevates urban heat island intensity.

GIS provides a spatial decision-support framework that can integrate heterogeneous data satellite imagery, terrain models, road networks, soil maps, census data into a unified analysis platform. When combined with

multi-criteria evaluation and land-change simulation models, GIS enables planners to visualise alternative development paths and their spatial consequences before committing to policy [5].

This study has three objectives:

- To quantify land-cover change in the kumasi metropolitan area between 2000 and 2020 using landsat satellite data;
- To identify zones with high and low development suitability through ahp-weighted overlay; and
- To project land-cover conditions in 2030 under three policy scenarios using a ca-markov model.

The results are intended to inform Kumasi's Spatial Development Framework, currently under revision by the Town and Country Planning Department.

II. LITERATURE REVIEW

A. GIS in Urban Planning

GIS entered urban planning practice in the late 1980s as a tool for zoning map production and infrastructure inventory management [6]. Its role has since expanded to include suitability analysis, impact assessment, and scenario modelling. Malczewski [7] reviewed GIS-based MCDA applications and found over 300 published studies between 1990 and 2010, with land suitability assessment as the most common application. Weighted overlay — assigning numerical weights to raster criterion layers and summing them — remains the most widely used spatial MCDA technique due to its conceptual simplicity and straightforward implementation in commercial GIS packages [5].

B. Multi-Criteria Decision Analysis and AHP

The Analytic Hierarchy Process, developed by Saaty [8], derives criterion weights from pairwise comparisons made by domain experts. Each pair of criteria is rated on a 1–9 scale indicating relative importance. The resulting comparison matrix is checked for consistency (consistency ratio CR < 0.10), and the principal eigenvector yields the final weights. AHP has been applied extensively to land-use suitability studies in tropical cities — for instance, Akinci et al. [9] used AHP-GIS to identify suitable sites for agricultural land in Turkey, and Duc [10] applied it to flood-risk zoning in Hanoi.

C. Land-Use Change Modelling

CA-Markov combines the spatial allocation logic of cellular automata with the transition probability estimation of Markov chains [11]. The Markov component calculates the probability of a cell transitioning from one land-cover class to another based on observed historical change rates. The CA component applies a spatial contiguity filter so that transitions preferentially occur adjacent to existing cells of the target class, producing spatially realistic growth patterns. Verburg et al. [17] reviewed land-use change modelling approaches and identified CA-Markov as particularly suited to data-scarce environments. Validation studies by Pontius et al. [12] and Mas et al. [13] have shown that CA-Markov captures gross quantity change well but can underperform on location accuracy when transition drivers are complex.

D. Remote Sensing for Land Cover Mapping

Landsat provides the longest continuous archive of moderate-resolution (30 m) Earth observation data, spanning four decades. Supervised classification using Maximum Likelihood (ML), Support Vector Machine (SVM), or Random Forest algorithms can map land cover with overall accuracies above 85% when training samples are representative and atmospheric correction is applied [14]. For change detection, post-classification comparison is the most common approach: independently classified maps are overlaid to identify cells that changed class between dates [15].

III. STUDY AREA AND DATA

A. Kumasi Metropolitan Area

Kumasi lies in the Ashanti Region of central Ghana (6°41'N, 1°37'W) at an elevation of approximately 270 m above sea level. The city occupies a gently undulating terrain dissected by the Subin, Aboabo, and Sisa rivers. Mean annual rainfall is 1,400 mm, distributed across two rainy seasons (March–July and September–November). The metropolitan area covers 262 km², and the surrounding peri-urban zone extends to roughly 500 km². Kumasi functions as the principal commercial and transport hub for the central forest zone of Ghana, connected to Accra (260 km south) and Tamale (380 km north) by trunk roads [2].

B. Data Sources

Satellite imagery was obtained from the USGS Earth Explorer platform: Landsat 5 TM (2000), Landsat 7 ETM+ (2010), and Landsat 8 OLI (2020). All scenes were acquired during the dry season (December–January) to minimise cloud cover and phenological variation. Atmospheric correction was performed using the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) module in ENVI 5.6. Ancillary data included: a 30-m Shuttle Radar Topography Mission (SRTM) DEM for slope derivation; road network vectors from OpenStreetMap (downloaded March 2023); soil association maps from the Ghana Soil Research Institute; and population census data (2000, 2010, 2021) from the Ghana Statistical Service [2].

C. Land Cover Classification

Five land-cover classes were defined: built-up (impervious surfaces, buildings), vegetation (forest, dense shrub), agricultural (farmland, plantations), water (rivers, ponds), and bare soil (exposed earth, construction sites). Training samples (150–250 pixels per class) were collected from high-resolution Google Earth imagery and field GPS points. The Maximum Likelihood classifier was applied in ArcGIS Pro 3.1. A 5 × 5 majority filter was applied post-classification to reduce salt-and-pepper noise [14].

D. Accuracy Assessment

A stratified random sample of 300 validation points (60 per class) was compared against visual interpretation of Google Earth imagery and field photographs. Accuracy metrics were computed from the confusion matrix. Pontius [19] recommended separating quantity error from location error when evaluating categorical map comparisons, a principle followed in this study.

Table 1. Classification accuracy assessment results

Year	Overall Accuracy (%)	Kappa Coefficient	Lowest Class Acc. (%)
2000	87.3	0.83	81.2 (Bare Soil)
2010	89.1	0.85	83.7 (Agricultural)
2020	91.4	0.88	85.4 (Bare Soil)

IV. METHODOLOGY

A. Land Cover Change Detection

Post-classification comparison was applied to the 2000–2010 and 2010–2020 image pairs. The area of each land-cover class was computed for each date, and a change matrix was generated identifying the from–to transitions between classes. Cells that changed from vegetation or agricultural to built-up were flagged as urbanisation conversions [15].

B. Suitability Analysis (MCDA-AHP)

Seven criteria were selected based on planning literature and local stakeholder consultation: (1) slope (derived from SRTM DEM); (2) distance to major roads; (3) distance to city centre; (4) soil drainage class; (5) flood risk zone (mapped from historical flood extent and DEM-based flow accumulation); (6) distance to water bodies; and (7) existing land use. Each criterion was reclassified to a common 1–5 suitability scale (1 = least suitable, 5 = most suitable) using thresholds from literature and local planning standards.

A panel of five experts (two urban planners, one hydrologist, one soil scientist, and one GIS analyst) performed pairwise comparisons. The resulting AHP weights are shown in Table 2.

Table 2. AHP criteria weights (consistency ratio CR = 0.064 < 0.10)

Criterion	Weight	Rank
Flood risk zone	0.261	1
Distance to roads	0.198	2
Slope	0.152	3
Distance to city centre	0.134	4
Soil drainage class	0.108	5
Existing land use	0.087	6
Distance to water bodies	0.060	7

Flood risk received the highest weight (0.261), reflecting Kumasi's recurrent flood hazard and the expert panel's prioritisation of risk avoidance over accessibility. Distance to roads ranked second (0.198) as a proxy for infrastructure access. The weighted overlay was computed as:

$$S = \sum(w_i \times c_i) \quad (1)$$

where S is the composite suitability score, w_i is the weight for criterion i, and c_i is the reclassified score (1–5). The resulting suitability map was reclassified into four zones: highly suitable (S > 4.0), moderately suitable (3.0–4.0), marginally suitable (2.0–3.0), and unsuitable (S < 2.0).

C. CA-Markov Simulation

Transition probability matrices were computed from the 2000–2010 and 2010–2020 change maps using the Markovian Transition Estimator in TerrSet 2020. A 5 × 5 contiguity filter defined the CA neighbourhood rule. The model was validated by simulating 2020 land cover from the 2000–2010 transition probabilities and comparing the result with the actual 2020 classification. Agreement was assessed using Kappa for location (K_location = 0.79), which falls within the acceptable range reported by Pontius et al. [12] for urban studies. Rimal et al. [20] and Hamad et al. [21] applied comparable CA-Markov frameworks to model urban expansion in South Asian and Middle Eastern cities respectively, achieving similar validation accuracy.

D. Scenario Analysis

Three scenarios were modelled for 2030:

- Business-as-Usual (BAU): Historical transition rates continue without policy intervention. All cells eligible for change.
- Planned Growth: Urban expansion is restricted to cells classified as 'highly suitable' or 'moderately suitable' by the AHP overlay. Cells in flood zones or on steep slopes (>15°) are locked as non-urban.
- Conservation: In addition to planned-growth constraints, a 200-m buffer around water bodies and all remaining vegetation patches larger than 5 hectares are designated as no-go zones.

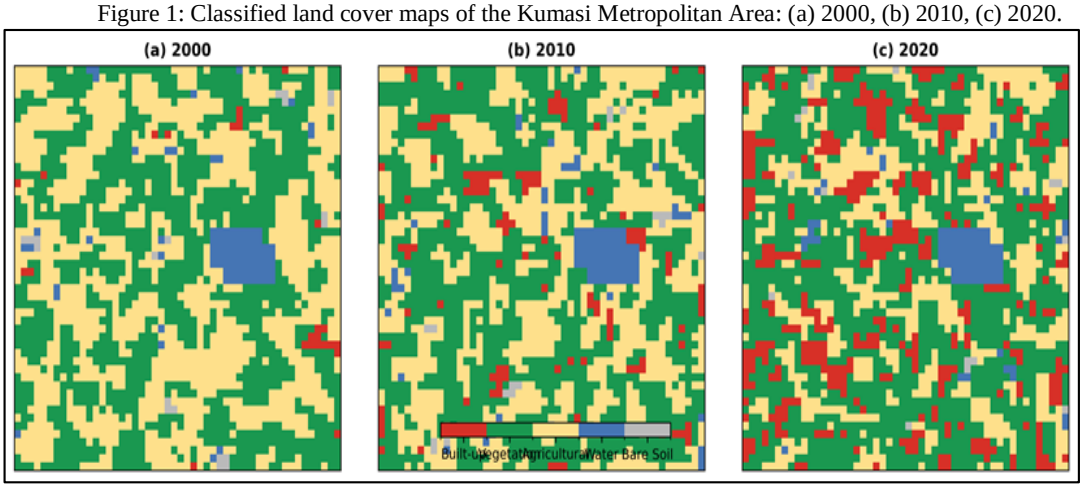
V. RESULTS AND DISCUSSION

Table 3. Land cover areas and net change, 2000–2020

Land Cover Class	2000 (km²)	2010 (km²)	2020 (km²)	Change 2000–2020 (km²)
Built-up	42.3	68.1	95.6	+53.3
Vegetation	98.7	78.4	60.2	−38.5
Agricultural	84.5	75.8	64.3	−20.2
Water	14.2	14.0	13.8	−0.4
Bare Soil	22.3	25.7	28.1	+5.8

Table 3 confirms a persistent urban expansion trend. Built-up area more than doubled from 42.3 km² in 2000 to 95.6 km² in 2020, an absolute gain of 53.3 km². This growth consumed 38.5 km² of vegetation and 20.2 km² of agricultural land. Water area remained essentially stable, losing only 0.4 km² — mostly through encroachment on pond margins. Bare soil increased by 5.8 km², representing active construction and cleared land awaiting development.

Figure 1: Classified land cover maps of the Kumasi Metropolitan Area: (a) 2000, (b) 2010, (c) 2020.



Note: Red = built-up, green = vegetation, yellow = agricultural, blue = water, grey = bare soil.

The spatial pattern of expansion (Figure. 1) shows growth radiating outward from the city centre along the main trunk roads — particularly the Accra road to the southeast and the Obuasi road to the southwest. Between 2010 and 2020, ribbon development along these corridors merged with previously isolated peri-urban settlements,

creating a continuous built-up fabric extending 12 km from the centre. Vegetation loss was most severe in the northern and eastern peri-urban zones, where cocoa farms were subdivided for residential plots. Abass et al. [18] documented similar peri-urban agricultural land loss in Kumasi over three decades using remote sensing.

Figure 2: Land cover area by class for 2000, 2010, and 2020.

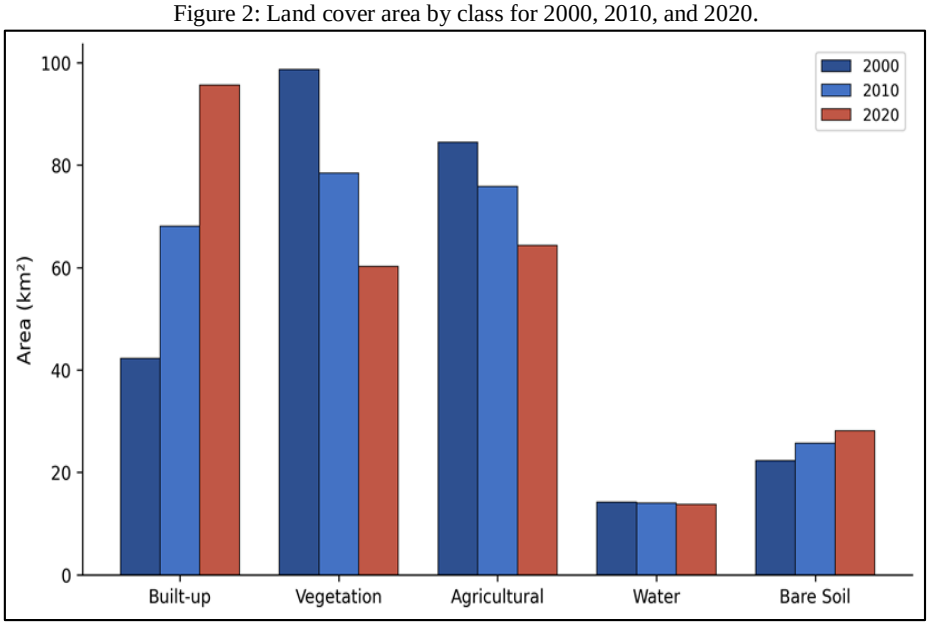
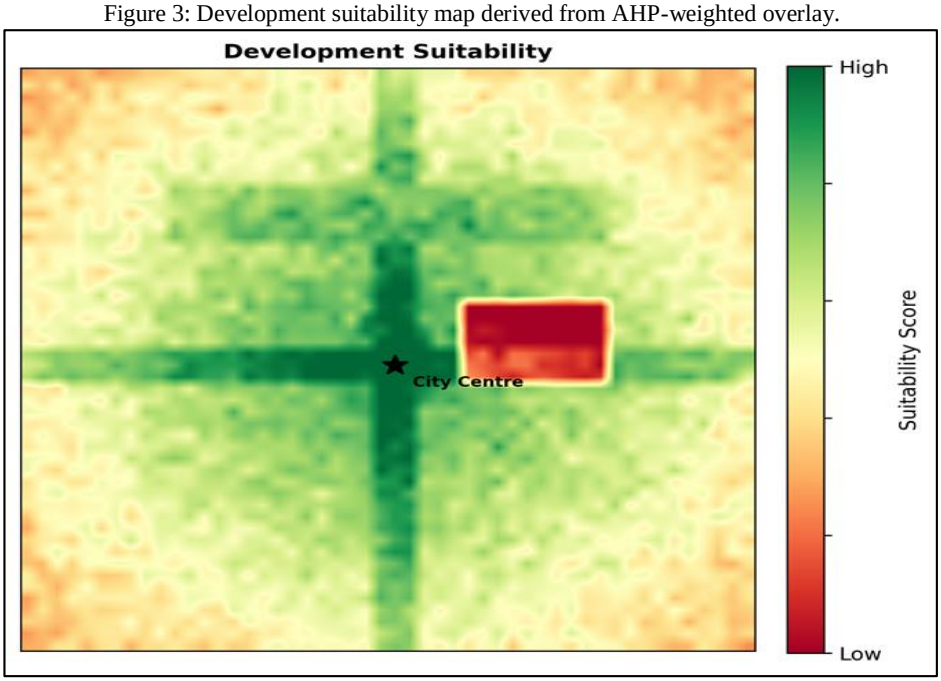


Figure 3: Development suitability map derived from AHP-weighted overlay.



Note: Green = highly suitable, red = unsuitable. Star marks the city centre.

The suitability map (Figure. 3) identifies a band of highly suitable land extending 3–5 km along the major road corridors, where slopes are gentle, infrastructure is accessible, and flood risk is low. Areas adjacent to the Subin and Aboabo river floodplains are classified as unsuitable despite their proximity to the centre — a direct consequence of the high weight assigned to flood risk (0.261). The southern peri-urban zone shows moderate suitability, constrained by steeper terrain and limited road access.

Table 4. Predicted land cover for 2030 under three scenarios

Scenario	Built-up 2030 (km²)	Increase from 2020 (%)	Agri. Land Lost (km²)
Business-as-Usual	131.9	38.0	18.4
Planned Growth	116.5	21.9	11.2
Conservation	104.2	9.0	5.8

Table 4 summarises the scenario projections. Under BAU, built-up area reaches 131.9 km² by 2030, consuming an additional 18.4 km² of agricultural land. The planned-growth scenario reduces this to 116.5 km² (+21.9%) by restricting development to suitable zones, saving 7.2 km² of farmland relative to BAU. The conservation scenario limits built-up growth to 104.2 km² (+9.0%) sufficient to house projected population growth at higher densities — while preserving 65% more agricultural land than BAU.

Figure 4: Built-up area in 2020 (actual) and projected 2030 under three growth scenarios. Percentage labels show increase from 2020.

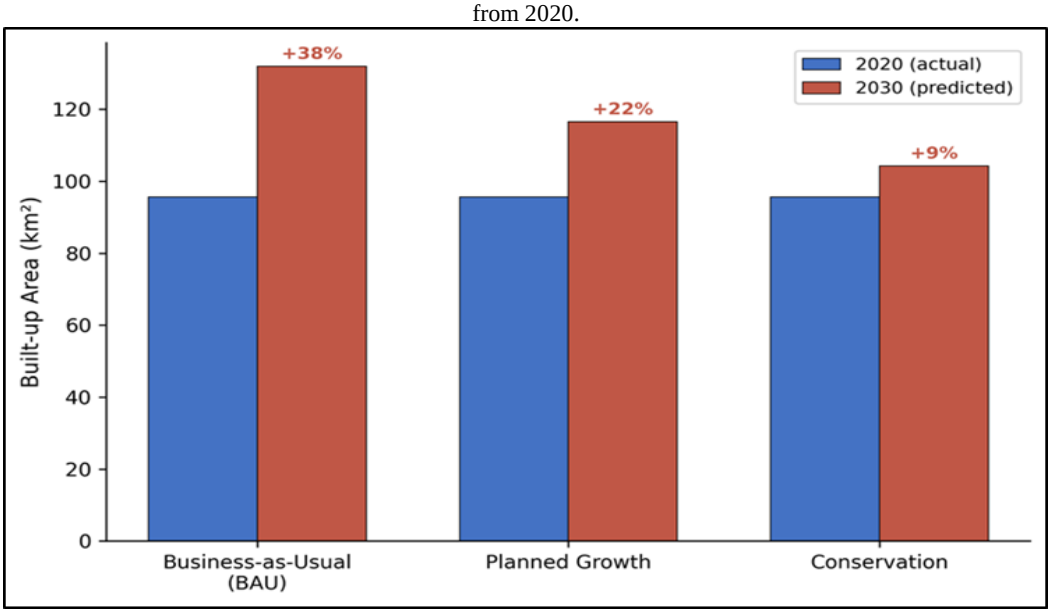
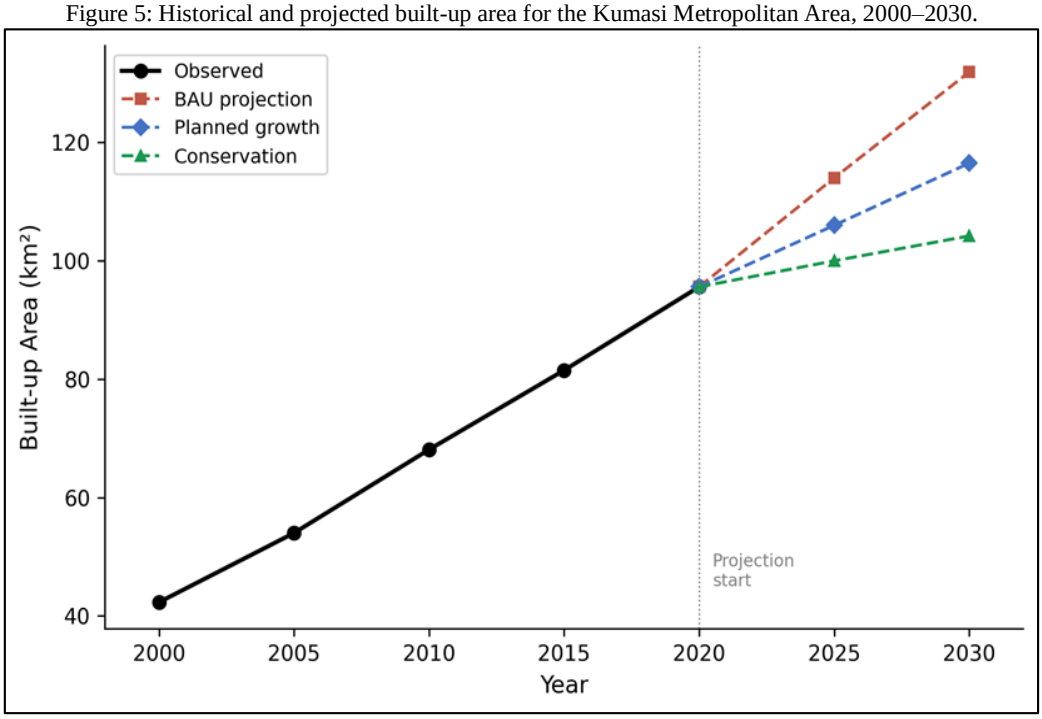


Figure 5: Historical and projected built-up area for the Kumasi Metropolitan Area, 2000–2030.



The trend analysis (Fig. 5) illustrates that Kumasi's built-up expansion has been approximately linear since 2000, adding roughly 2.7 km² per year. BAU extrapolates this trajectory. The planned-growth curve bends downward after 2020, reflecting the constraint that expansion is channelled into suitable zones rather than spreading uniformly. The conservation curve is the most conservative, implying densification policies — infill development, vertical construction, transit-oriented development that accommodate population growth within a smaller footprint.

From a planning perspective, the BAU scenario is unsustainable. It projects the loss of 28% of the remaining agricultural land within the metropolitan boundary, undermining local food supply chains and

livelihoods. Flood-prone areas would see further encroachment, increasing exposure of vulnerable populations. The planned-growth scenario offers a middle path: it accommodates 85% of BAU's spatial expansion while avoiding flood zones and steep slopes, directing growth to locations where road, water, and drainage infrastructure can be efficiently extended.

The conservation scenario demands the strongest policy commitment, including enforced green belts, minimum-density requirements in serviced areas, and active wetland restoration. While politically challenging, it aligns with Ghana's National Spatial Development Framework (2015–2035), which calls for compact, resilient urban forms [16].

VI. CONCLUSION

This study applied GIS-based MCDA and CA-Markov modelling to assess urban growth dynamics and evaluate future development pathways for Kumasi, Ghana. The principal findings are:

Built-up area more than doubled between 2000 and 2020, expanding from 42.3 km² to 95.6 km². The majority of this growth consumed vegetation (38.5 km² lost) and agricultural land (20.2 km² lost), concentrated along trunk road corridors radiating from the city centre.

AHP-weighted suitability analysis identified flood risk and road proximity as the two most influential criteria for development suitability (combined weight 0.459). Highly suitable zones form a corridor pattern along existing roads, confirming that infrastructure-led growth is the dominant spatial determinant.

Under business-as-usual, built-up area is projected to reach 131.9 km² by 2030 (+38%), consuming 18.4 km² of additional agricultural land. The planned-growth scenario reduces this expansion to 21.9% while preserving 65% more farmland by channelling development into high-suitability zones.

The conservation scenario limits expansion to 9% but requires densification policies that represent a significant departure from Kumasi's current low-density growth pattern.

These results provide a quantitative foundation for the ongoing revision of Kumasi's Spatial Development Framework. The suitability maps can be directly incorporated into zoning designations, and the scenario projections offer decision-makers a tangible basis for debating trade-offs between growth accommodation and environmental preservation. Future work will incorporate population projection data at the ward level, integrate transport accessibility modelling, and extend the analysis to secondary cities within the Ashanti Region. More advanced simulation tools such as the FLUS model [22] could improve location accuracy by coupling human and natural driving factors.

REFERENCES

[1] United Nations, Department of Economic and Social Affairs, *World Urbanization Prospects: The 2018 Revision*. New York, NY, USA: United Nations, 2019.

[2] Ghana Statistical Service, *2021 Population and Housing Census: General Report*. Accra, Ghana: Ghana Statistical Service, 2021.

[3] K. C. Seto, M. Fragkias, B. Guneralp, and M. K. Reilly, "A meta-analysis of global urban land expansion," *PLoS ONE*, vol. 6, no. 8, Art. no. e23777, Aug. 2011.

[4] A. K. Jha, R. Bloch, and J. Lamond, *Cities and Flooding: A Guide to Integrated Urban Flood Risk Management for the 21st Century*. Washington, DC, USA: World Bank, 2012.

[5] J. Malczewski, "GIS-based multicriteria decision analysis: A survey of the literature," *Int. J. Geogr. Inf. Sci.*, vol. 20, no. 7, pp. 703–726, Aug. 2006.

[6] Masser, *GIS Worlds: Creating Spatial Data Infrastructures*. Redlands, CA, USA: ESRI Press, 2005.

[7] Malczewski and C. Rinner, *Multicriteria Decision Analysis in Geographic Information Science*. Berlin, Germany: Springer, 2015.

[8] T. L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. New York, NY, USA: McGraw-Hill, 1980.

[9] H. Akinci, A. Y. Ozalp, and B. Turgut, "Agricultural land use suitability analysis using GIS and AHP technique," *Comput. Electron. Agric.*, vol. 97, pp. 71–82, Sep. 2013.

[10] T. T. Duc, "Using GIS and AHP technique for land-use suitability analysis," in *Proc. Int. Symp. Geoinformatics Spatial Infrastruct. Dev. Earth Allied Sci.*, Ho Chi Minh City, Vietnam, Nov. 2006.

[11] R. Eastman, *IDRISI Taiga Guide to GIS and Image Processing*. Worcester, MA, USA: Clark Labs, Clark University, 2009.

[12] R. G. Pontius Jr., E. Shusas, and M. McEachern, "Detecting important categorical land changes while accounting for persistence," *Agric. Ecosyst. Environ.*, vol. 101, nos. 2–3, pp. 251–268, Feb. 2004.

[13] J. F. Mas, M. Kolb, M. Paegelow, M. T. Camacho Olmedo, and T. Houet, "Inductive pattern-based land use/cover change models: A comparison of four software packages," *Environ. Modell. Softw.*, vol. 51, pp. 94–111, Jan. 2014.

[14] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sens. Environ.*, vol. 80, no. 1, pp. 185–201, Apr. 2002.

[15] Lu, P. Mausel, E. Brondizio, and E. Moran, "Change detection techniques," *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2401, Jun. 2004.
[16] Government of Ghana, *National Spatial Development Framework 2015–2035*. Accra, Ghana: Town and Country Planning Department, 2015.
[17] P. H. Verburg, P. P. Schot, M. J. Dijst, and A. Veldkamp, "Land use change modelling: Current practice and research priorities," *GeoJournal*, vol. 61, no. 4, pp. 309–324, Dec. 2004.
[18] Abass, S. K. Adanu, and S. Agyemang, "Peri-urbanisation and loss of arable land in Kumasi Metropolis in three decades: Evidence from remote sensing image analysis," *Land Use Policy*, vol. 72, pp. 470–479, Mar. 2018.
[19] R. G. Pontius Jr., "Quantification error versus location error in comparison of categorical maps," *Photogramm. Eng. Remote Sens.*, vol. 66, no. 8, pp. 1011–1016, Aug. 2000.
[20] B. Rimal, L. Zhang, H. Keshtkar, B. N. Haack, S. Rijal, and P. Zhang, "Land use/land cover dynamics and modeling of urban land expansion by the integration of cellular automata and Markov chain," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 4, Art. no. 154, Apr. 2018.
[21] R. Hamad, H. Balzter, and K. Kolo, "Predicting land use/land cover changes using a CA-Markov model under two different scenarios," *Sustainability*, vol. 10, no. 10, Art. no. 3421, Sep. 2018.
[22] X. Liu *et al*., "A future land use simulation model (FLUS) for simulating multiple land use scenarios by coupling human and natural effects," *Landscape Urban Plan.*, vol. 168, pp. 94–116, Dec. 2017.