Performance Evaluation of a Heterogeneous CPU-GPU Framework for Real-Time ECG Analytics

Leveraging OpenMP on CPUs and CUDA CNNs on NVIDIA A30 for Cardiac Signal Classification

Ariana Howell, La Chiara Landrum, Naw Safrin Sattar

Computer Science and Data Science Department, School of Applied Computational Sciences, Meharry Medical College, Nashville, Tennessee, USA

Email: ariana.howell@mmc.edu, llandrum24@mmc.edu, nawsafrin.sattar@mmc.edu

MOTIVATION

- Digital twins depend on rapid physiological data processing for accurate patient modeling[1]
- Real-time cardiac monitoring requires extremely low processing latency [2]
- Heterogeneous HPC combining multicore CPUs and GPUs can accelerate clinical AI pipelines [3],[4].

PROBLEM STATEMENT

Current research on AI-enabled ECG analytics largely emphasizes predictive performance [1] or isolated hardware acceleration [4], yet there remains a methodological gap in assessing whether these pipelines are suitable for deployment within real-time digital twin environments [2]. Existing studies rarely integrate AI model optimization, heterogeneous computing evaluation, and scalability analysis into a unified workflow capable of assessing overall digital twin readiness [3].

CONTRIBUTIONS

- **A proposed workflow architecture** for evaluating AI-enabled ECG analytics pipelines intended for digital twin deployment.
- **An integrated evaluation methodology** that combines predictive performance, computational efficiency, and scalability within a single assessment framework.
- **A digital twin readiness assessment** weak scaling, and resource utilization. **framework** based on measurable criteria such as model performance, latency, throughput, strong scaling.

PROPOSED WORKFLOW

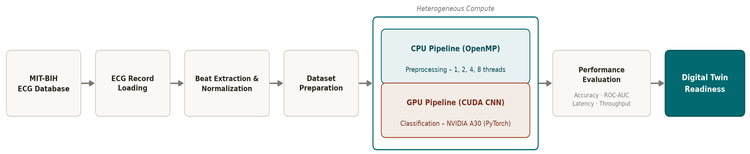


Figure 1. Heterogeneous CPU–GPU ECG analytics workflow. MIT-BIH ECG data undergo preprocessing and beat extraction using OpenMP, followed by GPU-accelerated CNN classification and performance evaluation.

PARALLEL IMPLEMENTATION

CPU: OpenMP

- Sequential baseline established
- ECG preprocessing parallelized using OpenMP
- Tested with 1 ,2, 4, and 8 CPU threads
- Static workload distribution; Runtime averaged over 10 repeated runs

GPU: CUDA

- PyTorch CNN Trained on NIVIDIA A30 GPU
- Five-Class ECG beat classification (N, S, V, F, Q)
- CUDA used for training and inference
- Accuracy, ROC-AUC, confusion matrix and latency evaluated

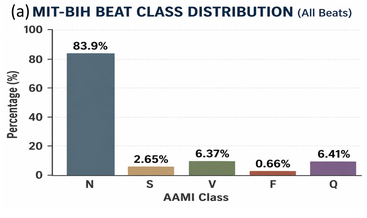
DATA & EXPERIMENTAL SETUP

Table 1. ECG Dataset Configuration Summarizes the MIT-BIH Arrhythmia Database, record count, sampling frequency, total beats, and AAMI heartbeat classes.

Component	Configuration
Dataset	MIT-BIH Arrhythmia Database
Records	48 two-lead ECG Records
Sampling Frequency	360 Hz
Total Beats	125,556
Classes	N,S,V,F,Q (AAMI EC57)

Table 2. Heterogeneous CPU–GPU Experimental Configuration Summarizes the CPU, OpenMP thread counts, GPU, and software frameworks used for performance evaluation.

Component	Configuration
CPU	Intel Xeon Gold 624BR (24 Cores)
CPU Threads	1,2,4,8
CPU Framework	OpenMP (C++)
GPU	NVIDIA A30
GPU Framework	PyTorch 2.1.3/CUDA 13.0



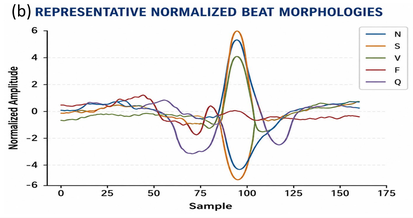
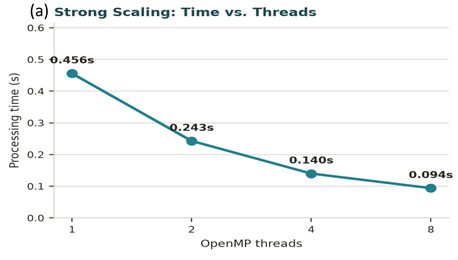


Figure 2. (a) MIT-BIH arrhythmia class distribution. Distribution of ECG beats across the five AAMI heartbeat classes (N, S, V, F, and Q), illustrating substantial class imbalance.
Figure 2. (b) Representative ECG beat morphologies. Normalized heartbeat waveforms for the five AAMI classes, illustrating differences in temporal morphology.

RESULTS: CPU STRONG SCALING



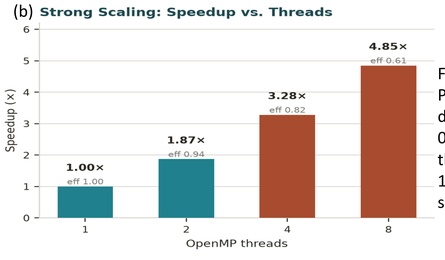


Figure 3(a) and Figure 3(b.) Preprocessing time decreased from 0.456 s to 0.094 s as the OpenMP thread count increased from 1 to 8, achieving a 4.85× speedup.

RESULTS: CPU WEAK SCALING

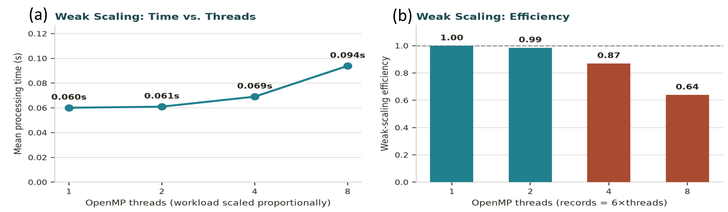


Figure 4(a) and Figure 4(b.) OpenMP weak scaling performance. Execution time and parallel efficiency across increasing thread counts, illustrating the framework's ability to process larger workloads. Efficiency declined from 1.00 to 0.64 at 8 threads, consistent with bandwidth and synchronization overhead.

CPU vs GPU INFERENCE THROUGHPUT

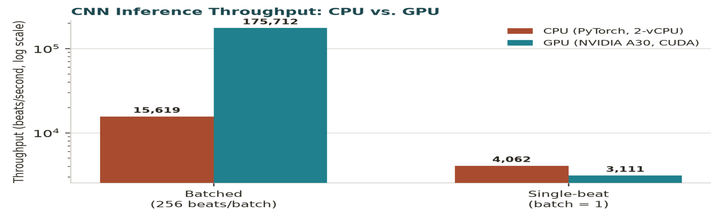


Figure 5. Batched GPU inference reached **175,712** beats/s, approximately **11.2×** the throughput of batched CPU inference.

CLASSIFICATION RESULTS

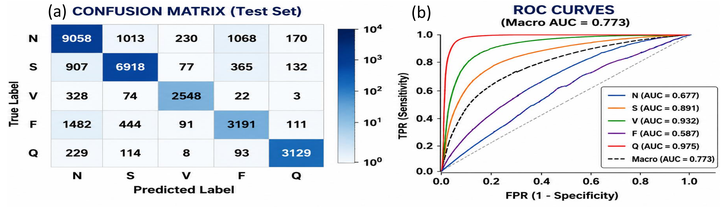


Figure 6(a) and Figure 6(b). The CNN achieved 78.17% accuracy and a macro ROC-AUC of 0.773, with weaker discrimination for the S and F beat classes.

DIGITAL TWIN READINESS SCORECARD

CAPABILITY	FINDING	READINESS
CPU ACCELERATION	**4.85×** Speedup	READY
PARALLEL SCALING	**81%** Efficiency	READY
CNN ACCURACY	**78.17%**	PARTIAL
MINORITY ARRHYTHMIAS	Needs Improvement	LIMITATION
GPU INFERENCE	**175,712** beats/s	READY
LATENCY	**5.7** µs/beat	READY
OVERALL DIGITAL TWIN READINESS	**Suitable for Real-Time Deployment**	READY

Table 3. Digital twin readiness assessment. Evaluation of the framework's computational performance, clinical classification capability, and deployment readiness against selected criteria.

FUTURE WORK

- **Streaming Real-Time ECG Analytics** for continuous patient monitoring [2].
- **Advanced CNN Optimization** to improve inference latency and energy efficiency
- **Full 24-core CPU scaling**
- **Multi-GPU Heterogeneous Scaling** for distributed deep learning acceleration.
- **Multimodal Digital Twin Integration** combining ECG with complementary physiological signals.
- **Edge-to-Cloud Deployment** enabling scalable, real-time digital twin healthcare systems [3].

CONCLUSIONS

This research presented a heterogeneous high-performance computing framework integrating multicore CPU parallelization with GPU-accelerated deep learning for ECG analytics. By combining predictive evaluation with computational benchmarking, the proposed Digital Twin Readiness Assessment Framework provides a comprehensive methodology for evaluating future AI-enabled healthcare systems. The framework establishes a foundation for future digital twin research while demonstrating how heterogeneous computing can accelerate biomedical analytics beyond traditional machine learning evaluation.

REFERENCES

[1] Y. Wu and B. Ghoraani, "Biological Signal Processing and Analysis for Healthcare Monitoring," Sensors, vol. 22, no. 14, p. 5341, 2022. doi:10.3390/s22145341.
[2] L. M. Pinho, *Real-Time Parallel Programming: State of Play and Open Issues*. arXiv:2303.11018, Mar. 2023. doi:10.48550/arXiv.2303.11018.
[3] A. Kanduri, S. Shahhosseini, E. K. Naeini, H. Alikhani, P. Liljeberg, N. Dutt, and A. M. Rahmani, *Edge-centric Optimization of Multi-modal ML-driven eHealth Applications*. arXiv:2208.02597, 2022. doi:10.48550/arXiv.2208.02597.
[4] X. Dong, C. Chen, Q. Geng, W. Zhang, and X. D. Zhang, "Fast Algorithm Based on Parallel Computing for Sample Entropy Calculation," *IEEE Access*, vol. 9, pp. 20223–20234, 2021. doi:10.1109/ACCESS.2021.3054750.

ACKNOWLEDGEMENTS

This work utilized ACES advanced computing resources at Texas A&M High Performance Research Computing through ACCESS allocation CIS260987, supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296. The authors acknowledge NSF Award No. 2401928 , Meharry Medical College, Fisk University, and the MS-CC Undergraduate Summer Research Internship Program. Special thanks to Dr. Uttam Ghosh, Dr. Pushpita Chatterjee, Dr. Lubna Pinky, Dr. Debashis Das, Opeyemi Omotosho for their research mentorship and Dr. Naw Safrin Sattar for her instructional guidance.