Green CRISP-ML(Q)

Embedding Sustainable AI Development Through Comprehensive Environmental Profiling

Vaibhav Saran, Dr. Asif Imran

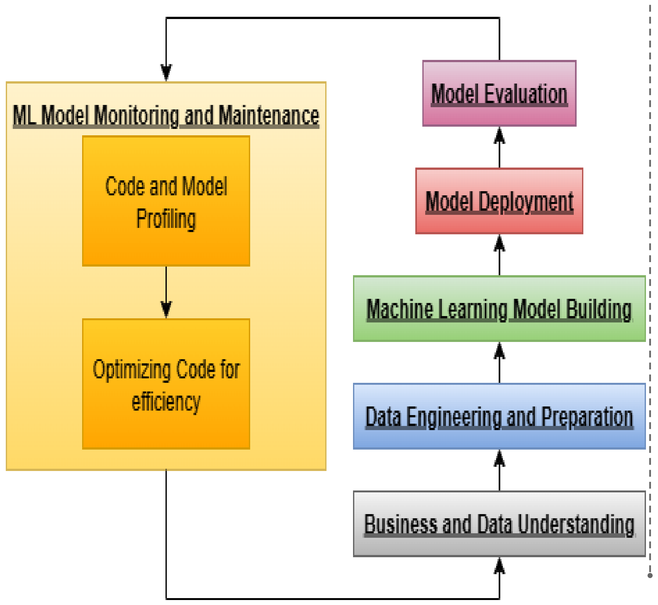
Introduction

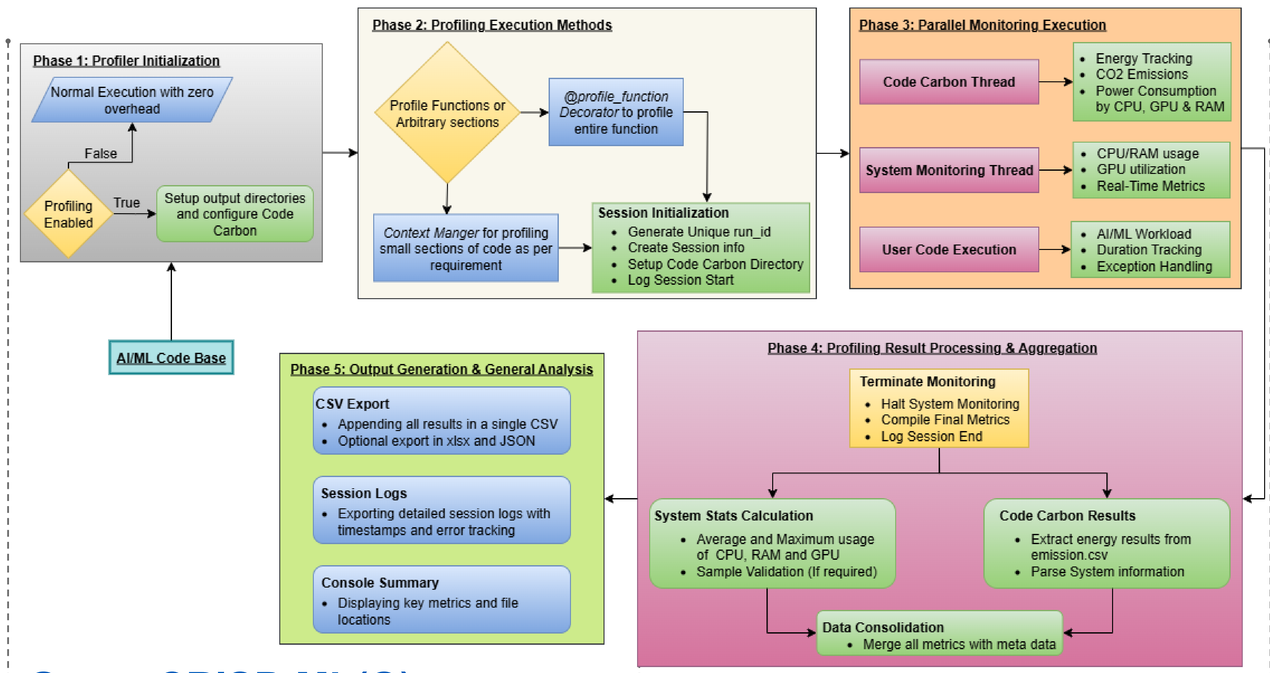
Training advanced AI systems requires substantial energy, such as the 1.287 GWh needed for a single ChatGPT training run, equivalent to the annual electricity consumption of approximately 120 American households. As AI continues to permeate all sectors, the urgent need for sustainable development practices has become critical.

Background

There are a lot of per-existing architectures for robust AI development:

- **CRISP-ML(Q)**:The Cross-Industry Standard Process for Machine Learning with Quality Assurance extends traditional CRISP-DM to address unique ML development challenges through six phases: Business Understanding, Data Engineering, Model Building, Model Evaluation, Model Deployment, and Model Monitoring [1].
- IBM **ASUM-DM** (Analytics Solution Unified Method – Data Mining): A framework presented by IBM for their lifecycle of AI Projects and development incorporating agile lifecycle [2].
- Microsoft **TDSP** (Team Data Science Process): This framework was given by Microsoft which is used by their AI team to work on various projects.





Green CRISP-ML(Q)

Green CRISP-ML(Q) extends CRISP-ML(Q) by integrating sustainability checkpoints throughout the AI development lifecycle, enabling practitioners to make informed environmental trade-offs without disrupting existing workflows. These checkpoints seamlessly integrate into existing workflows without disrupting established development processes:

A. **Business & Data Understanding**: Establish carbon budgets alongside performance targets

B. **Data Engineering & Preparation**: Monitor energy consumption during preprocessing and feature engineering.

C. **Model Building & Training**: Real-time energy monitoring with GPU optimization recommendations.

D. **Model Evaluation**: Include sustainability metrics alongside accuracy measures.

E. **Model Deployment**: Energy-efficient configuration and infrastructure optimization.

F. **Monitoring & Maintenance**: Continuous production monitoring with automated sustainability alerts.

Profiler Architecture

We implement a 5-phase architecture using combination of Scalene [3] and CodeCarbon [4] that provides comprehensive monitoring with minimal overhead through intelligent initialization and parallel execution.

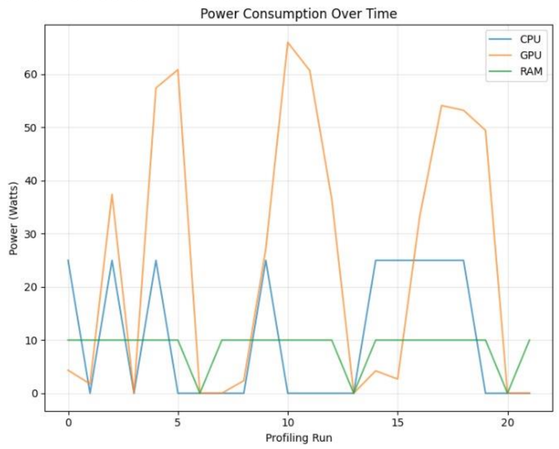
Phase 1: Profiler Initialization: Binary execution mode with zero overhead when disabled, full monitoring when enabled.

Phase 2: Profiling Execution Methods: *Function Decorator*: @profile_function and context managers for flexible code profiling with automatic session management.

Phase 3: Parallel Monitoring Execution: Three concurrent threads: CodeCarbon (energy/CO2 tracking), System Monitoring (CPU/GPU/RAM metrics), and User Code Execution.

Phase 4: Result Processing & Aggregation: Consolidates all monitoring data with comprehensive statistics and metadata.

Phase 5: Output Generation & Analysis: Exports 43-column datasets, session logs, and real-time performance summaries.



Experiments and Conclusion

Dataset & Model: Neural Collaborative Filtering (NCF) on MovieLens-20M dataset

Environment: Intel Xeon Gold 6126 (48 cores), Quadro RTX 6000 GPU, 187.4GB RAM

Training Dominance: Model training phases account for 93.5% of total CO2 emissions (114.60g out of 122.62g), demonstrating the critical need for training optimization

GPU Underutilization: Average GPU utilization of only 8.57% during training reveals significant optimization potential through improved acceleration and larger batch sizes

Resource Efficiency: Total energy consumption of 0.333 kWh with efficiency rate of 0.0357 kWh/hour indicates relatively efficient operation with room for improvement

Low CPU Utilization: CPU usage ranged from 0.52% to 4.14%, suggesting substantial opportunities for parallel processing optimization

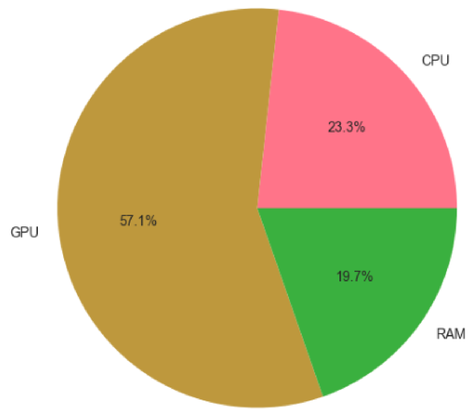
Future Directions

Future work includes expanding validation across diverse AI domains and developing automated optimization recommendations for sustainable AI development at scale.

```
============================================================
POWER CONSUMPTION ANALYSIS
============================================================

Average Power Consumption:
   CPU Power (W)  GPU Power (W)  RAM Power (W)  Total Power (W)
0          10.23          25.06           8.64            43.93
```

Average Power Distribution by Component



References

[1] https://arxiv.org/pdf/2003.05155
[2] Analytics Solutions Unified Method [IBM-ASUM-DM]
[3] https://github.com/plasma-umass/scalene
[4] https://codecarbon.io/
[5] https://arxiv.org/pdf/1708.05031

University at Buffalo The State University of New York

Engineering Science MS AI Program
vsaran@buffalo.edu
asifimra@buffalo.edu