ECE361E
Homework 2 - Complete Solutions

Edge Computing Systems

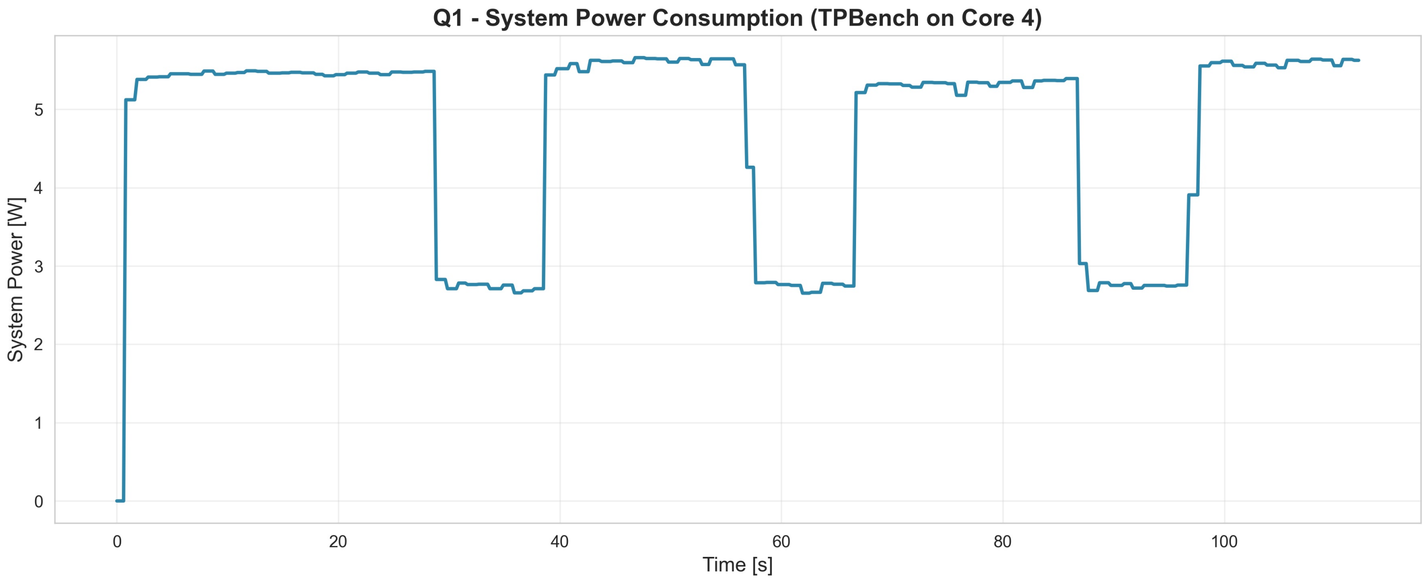
Names: Resul Ovezov, Ryane Li

Resul: Completed Problems 1 and 2
Ryane: Completed Problem 3

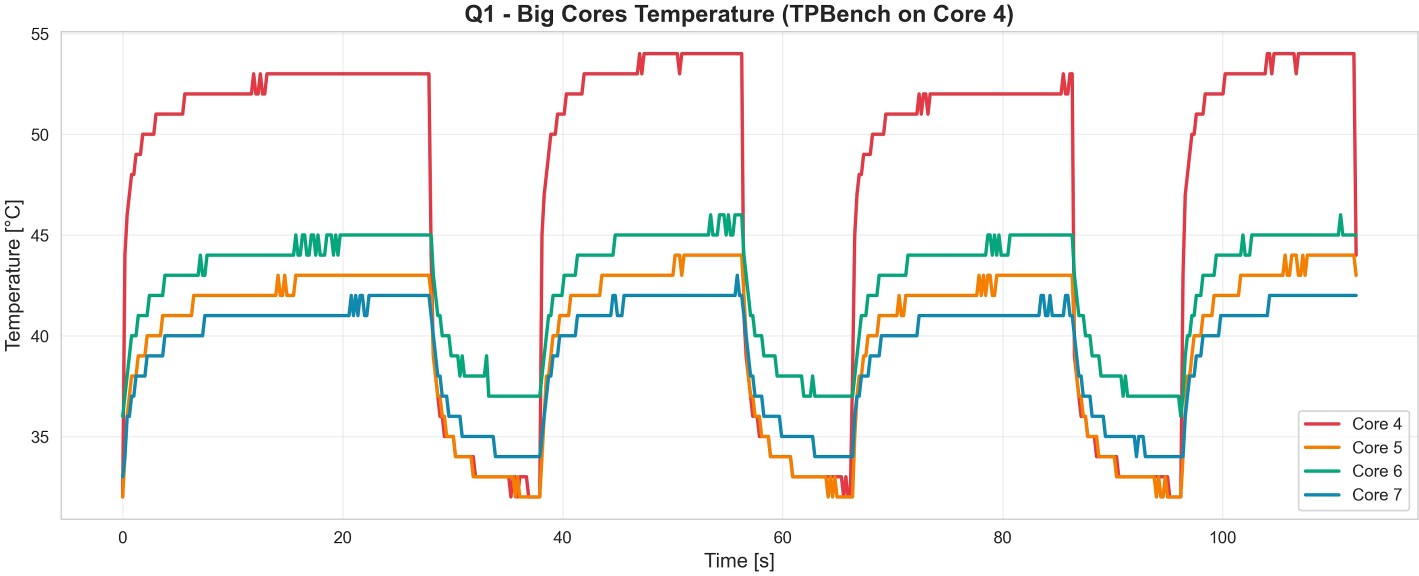
Problem 1: Odroid MC1 Benchmark Measurements

Question 1: TPBench Measurement [12p]

Plot 1: System Power Consumption

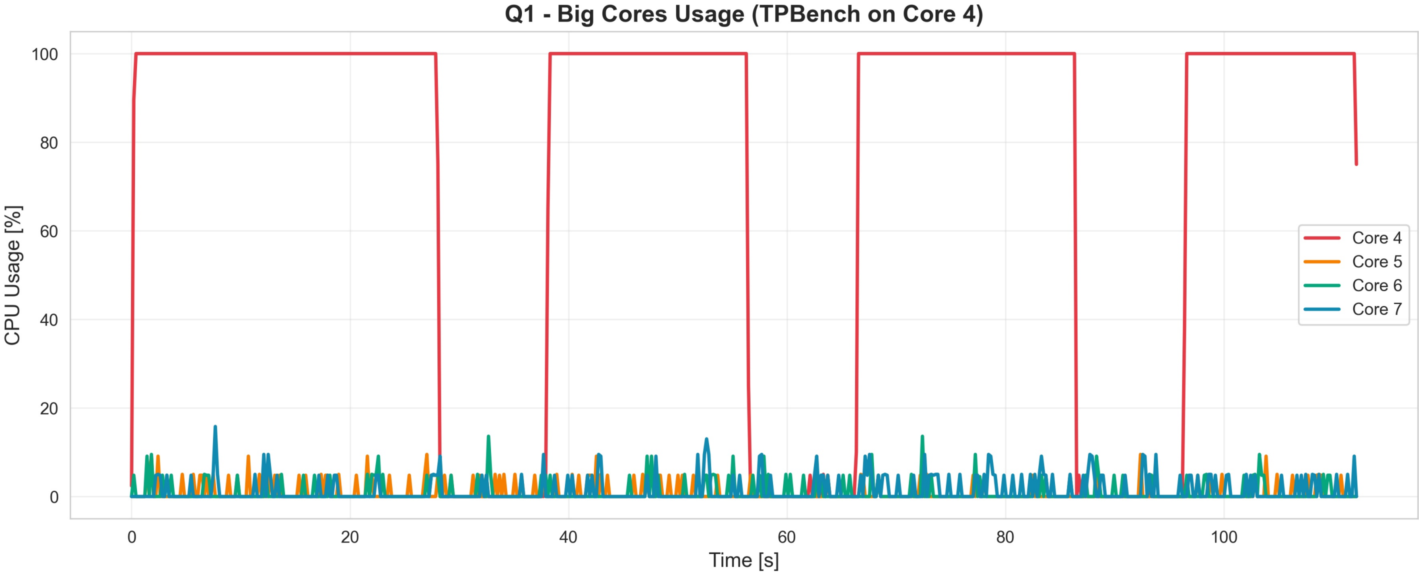


Plot 2: Big Cores Temperature



ECE361E - Homework 2 Solutions

Plot 3: Big Cores Usage



Question 2: Benchmark Phase Identification [3p]

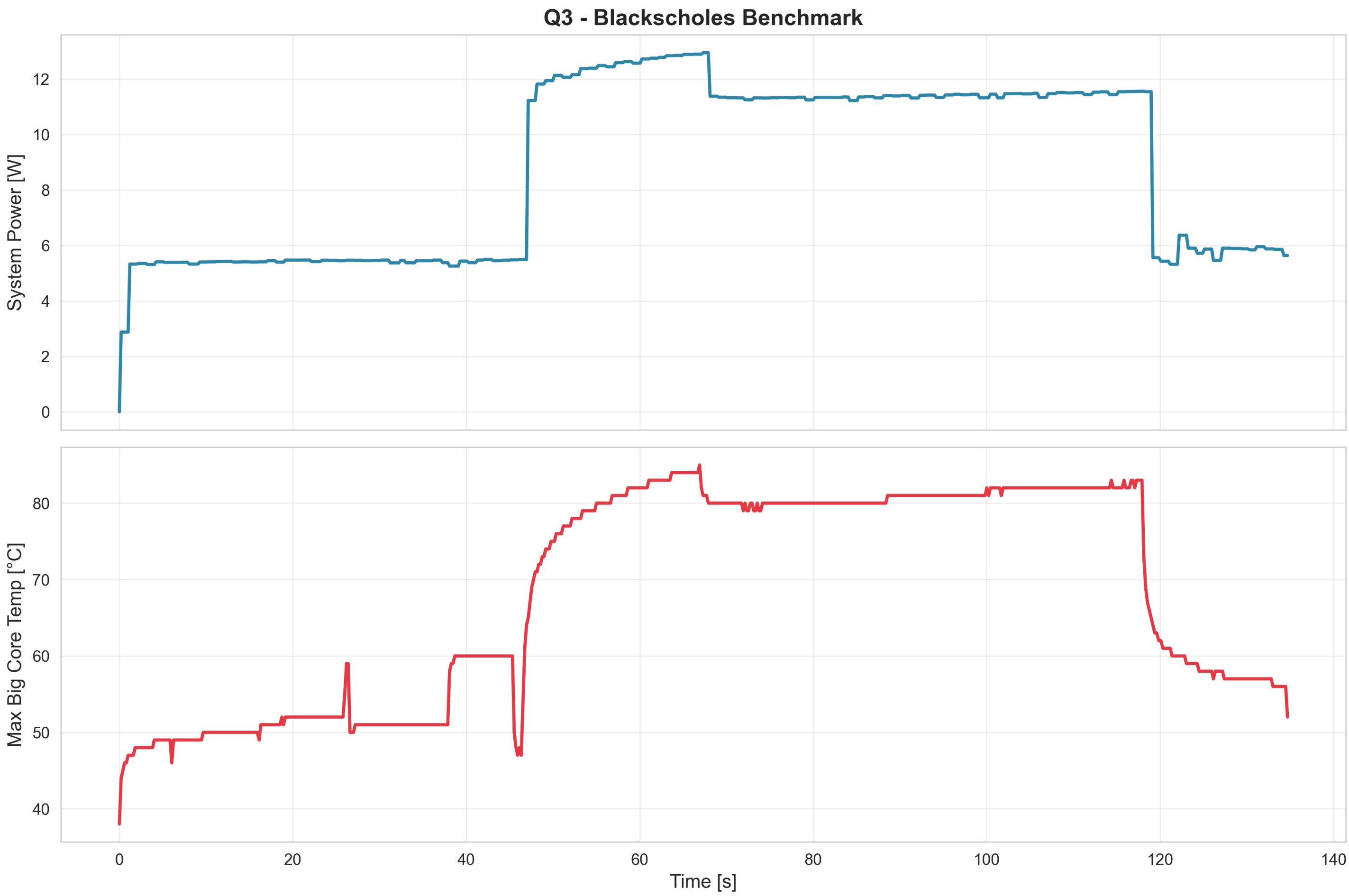
Question: How many phases of benchmark execution can you identify based on the temperature variation in the plot? A phase is a significant increase in the temperature over an extended period of time.

Answer:

Based on the temperature variation in the TPBench temperature plot, 4 distinct phases can be identified.

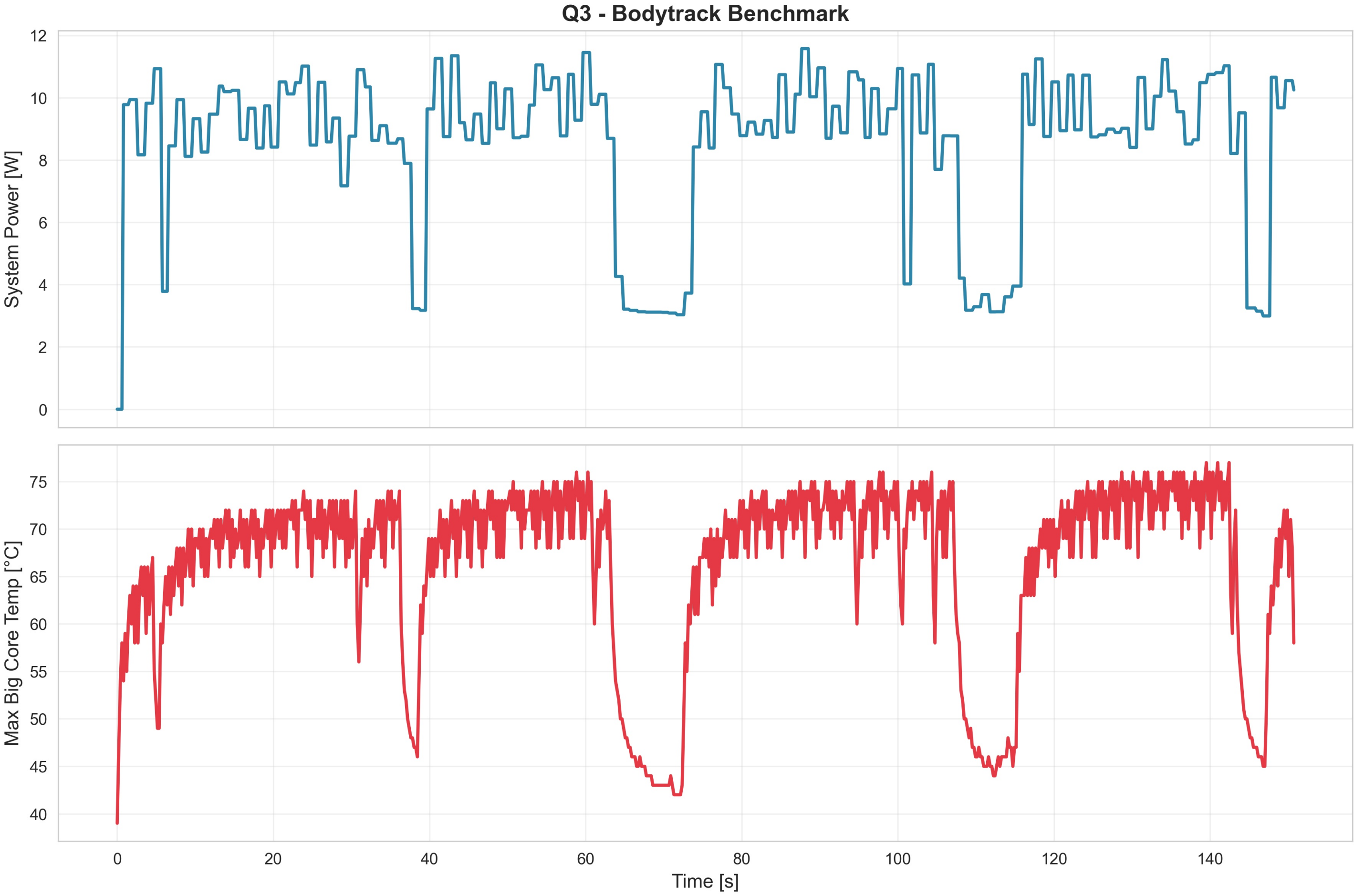
Question 3: Blackscholes and Bodytrack Benchmarks [15p]

Blackscholes Benchmark Plots



ECE361E - Homework 2 Solutions

Bodytrack Benchmark Plots



ECE361E - Homework 2 Solutions

Table 1: Benchmark Metrics

Benchmark	Run time [s]	Avg. power [W]	Avg. max temp [°C]	Max temp [°C]	Energy [J]
blackscholes	134.6790	8.7833	67.7806	85.0000	1182.9197
bodytrack	150.7130	8.5200	66.2176	77.0000	1284.0773

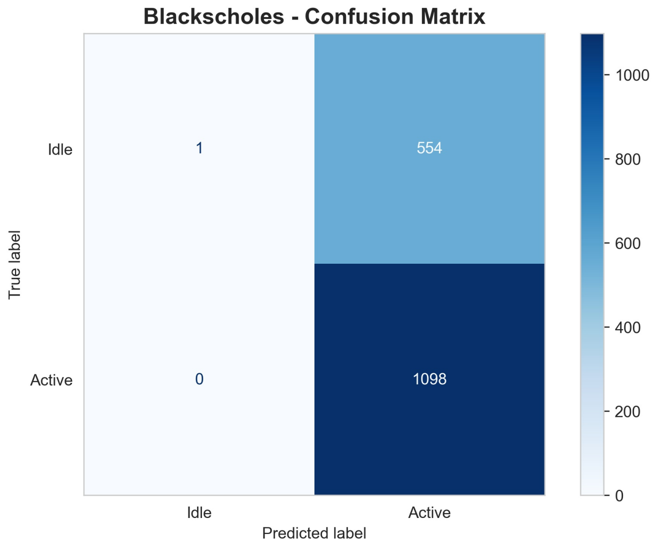
Problem 2: System Power Prediction

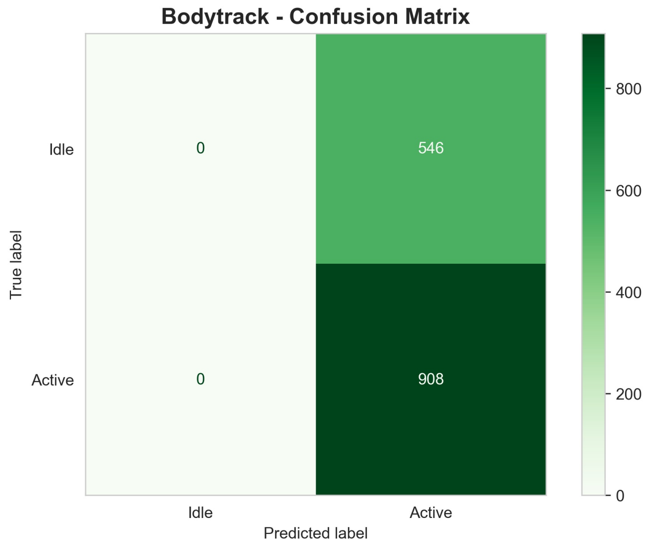
Question 1: SVM Classification [20p]

Table 2: SVM Classification Performance

Benchmark	Accuracy [%]	Avg. Precision	Avg. Recall	Avg. F1-Score
blackscholes	66.4852	0.8323	0.5009	0.4011
bodytrack	62.4484	0.3122	0.5000	0.3844

Confusion Matrices





Performance Analysis:

The performance is terrible. The model just ends up predicting everything as active and does nothing else on the testing set even though it is accurate on the training set and is able to differentiate. Features were normalized but still the performance is terrible.

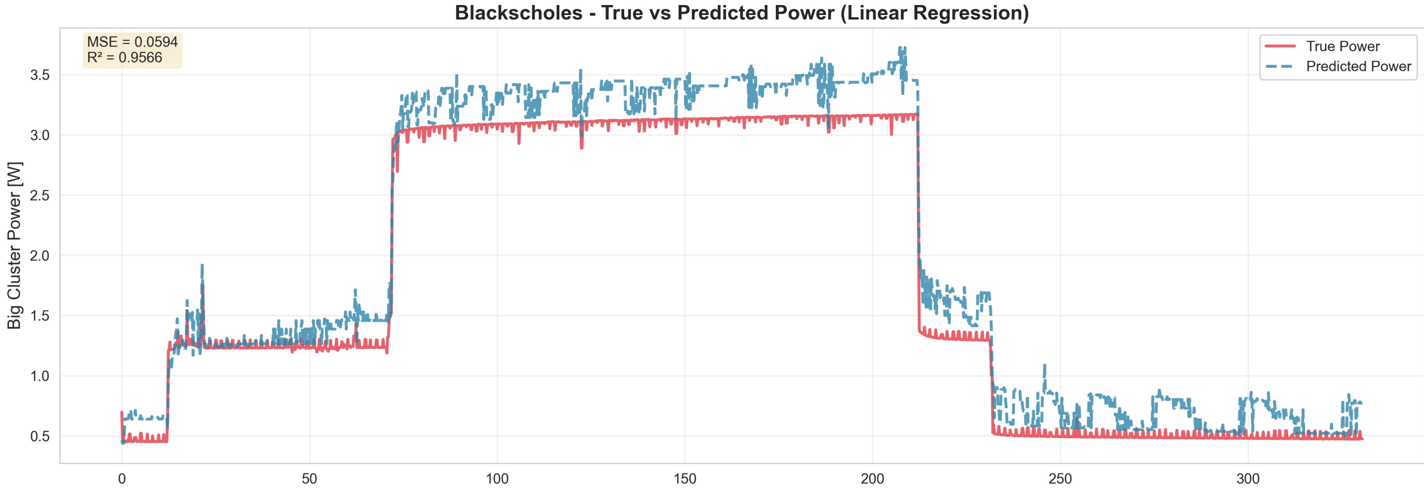
Question 2: Linear Regression for Power Prediction [5p]

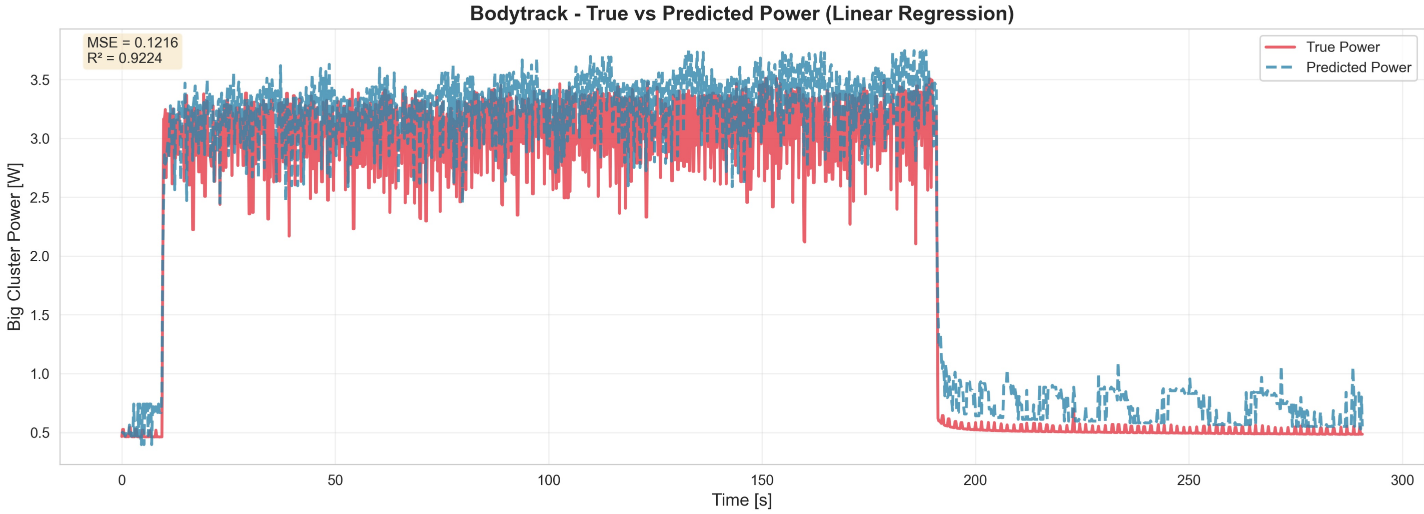
Table 3: Linear Regression Performance

Dataset	training	blackscholes	bodytrack
R^2	0.9870	0.9566	0.9224
MSE	0.0102	0.0594	0.1216

ECE361E - Homework 2 Solutions

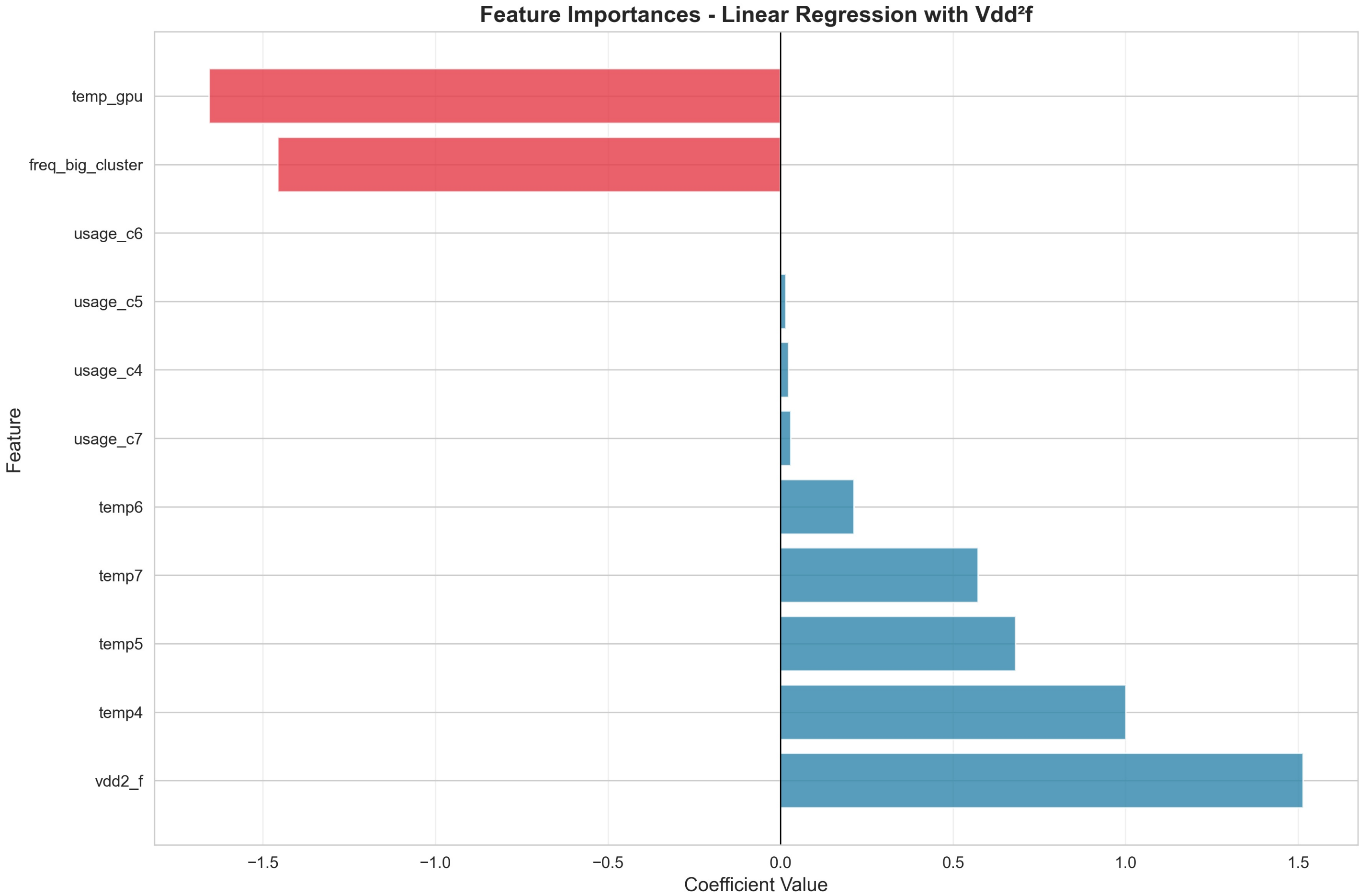
Power Prediction Results





Question 3: Feature Engineering with Vdd^2*f [15p]

Feature Importance Analysis



ECE361E - Homework 2 Solutions

Top 3 Positive Features Contributing to Big Cluster Power:

1. vdd2_f (Vdd^2*f) [1.514170]:

This feature based on the dynamic power formula P proportional to V^2*f has the strongest positive correlation with power consumption, confirming the theoretical relationship between voltage, frequency, and dynamic power.

2. temp4 [1.001162]:

Temperature of Core 4. Most likely the core most frequently assigned by the OS to take on tasks. Assuming that is the case, more tasks lead to more power consumption which generates more heat.

3. temp5 [0.681568]:

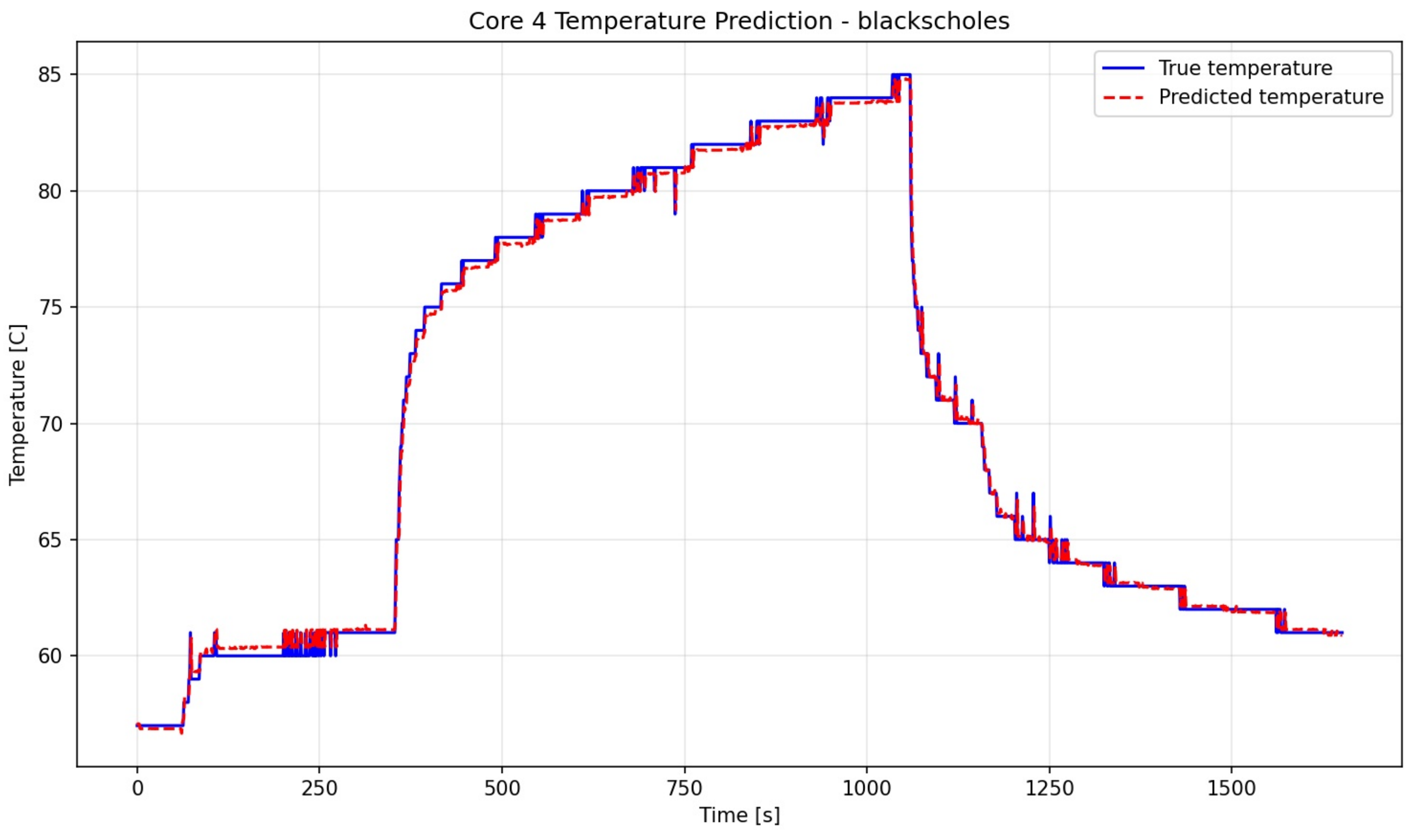
Temperature of Core 5. Most likely the core second most frequently assigned by the OS to take on tasks, although less by a decent margin.

Note: The feature importance might be different depending on how parallelizable the task is that's running.

Problem 3: Temperature Prediction using Neural Networks

Question 1: MLPRegressor for Temperature Prediction

Blackscholes - Core 4 Temperature Prediction



Bodytrack - Core 4 Temperature Prediction

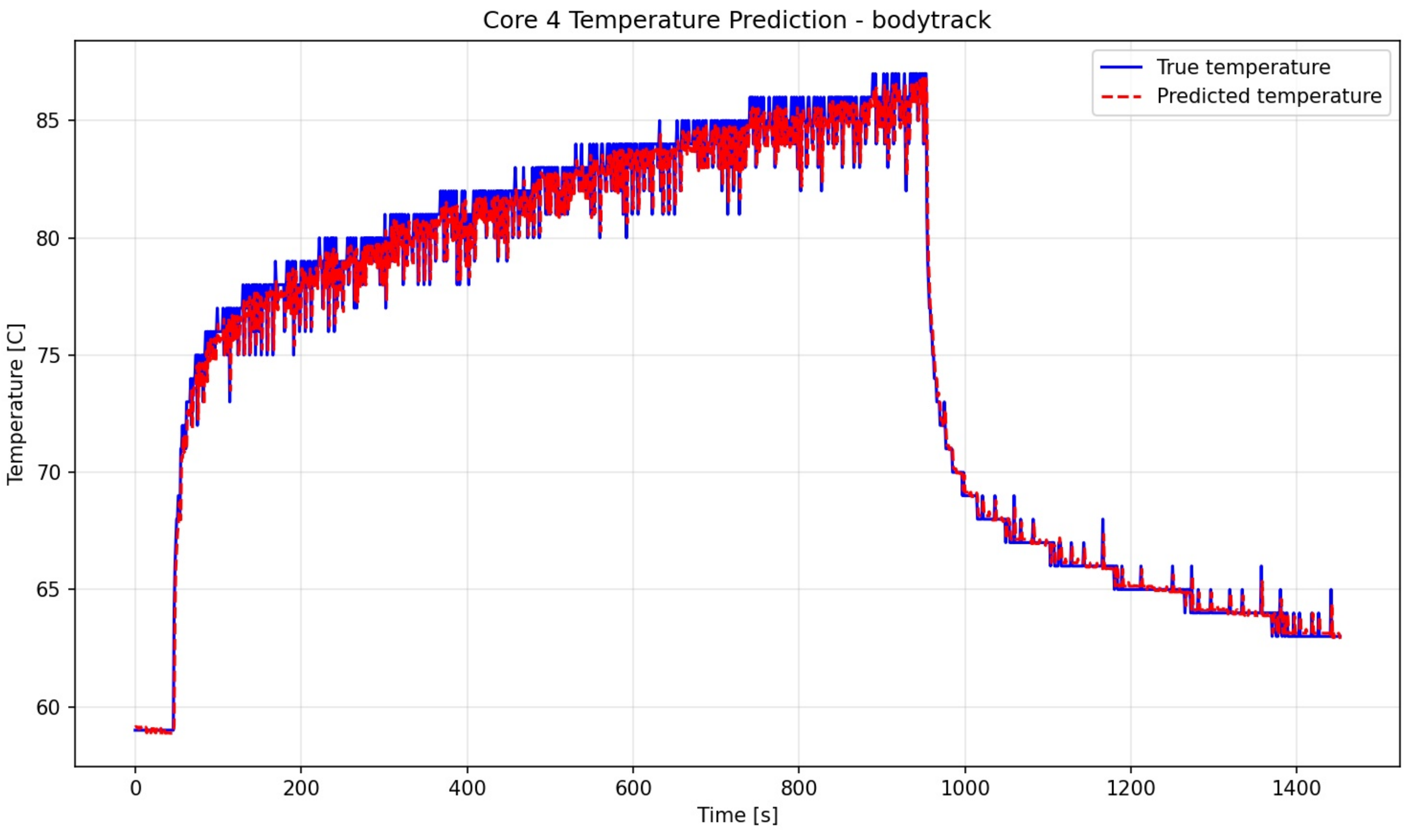


Table 5: Test MSE Results for All Cores

Dataset	Test MSE (Core 4)	Test MSE (Core 5)	Test MSE (Core 6)	Test MSE (Core 7)
blackscholes	0.156752	0.153057	0.326226	0.114536
bodytrack	1.165708	1.117478	1.392798	0.661181

ECE361E - Homework 2 Solutions

Question 2: Techniques to Improve Regressor Performance

Two or more techniques that can improve the temperature prediction model:

1. Feature scaling/normalization:

Apply StandardScaler or MinMaxScaler to non-temperature features (e.g., usage_c4-usage_c7, freq_big_cluster). Scaling puts predictors on a similar scale so the MLP converges faster and often generalizes better.

2. Temporal features:

Add lagged temperatures (e.g., from t-2, t-3) or rolling statistics (moving average, standard deviation) as features. This helps capture dynamics and thermal inertia.

BONUS Question 3: On-Demand Governor Analysis [10Bp]

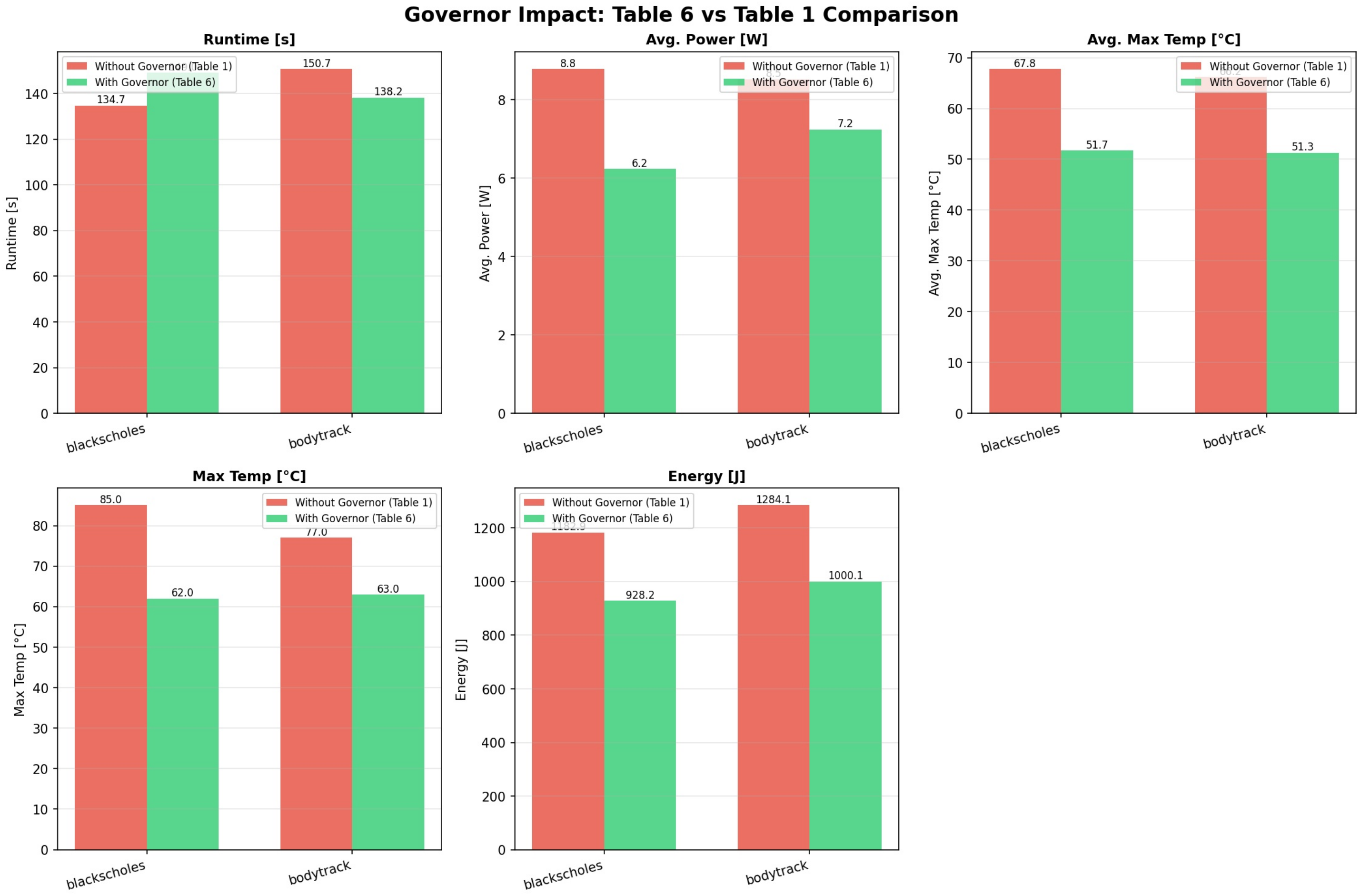
Using an Odroid MC1, we already implemented an on-demand governor algorithm. Table 6 shows the results obtained for both bodytrack and blackscholes benchmarks when executed with the on-demand governor active and a temperature threshold of 60C.

What are the possible "cyber-physical" trade-offs when having such a governor running? Discuss such trade-offs by comparing the runtime, average power consumption, thermal limits, and energy consumption of each benchmark with what you already obtained in Table 1.

Table 6: On-Demand Governor Results (60C threshold)

Benchmark	Runtime [s]	Avg. power [W]	Avg max temp [C]	Max temp [C]	Energy [J]
blackscholes	148.99	6.23	51.71	62	928.23
bodytrack	138.20	7.24	51.32	63	1000.12

Governor Impact Comparison



Answer: Cyber-Physical Trade-offs Analysis

Quantitative Comparison (Table 6 vs Table 1):

blackscholes:

- Runtime: 134.68s -> 148.99s (+10.6% slower)
- Avg. power: 8.78W -> 6.23W (-29.1% reduction)
- Avg. max temp: 67.78C -> 51.71C (-23.7% reduction)
- Max temp: 85C -> 62C (-27.1% reduction)
- Energy: 1182.92J -> 928.23J (-21.5% reduction)

bodytrack:

- Runtime: 150.71s -> 138.20s (-8.3% faster)
- Avg. power: 8.52W -> 7.24W (-15.0% reduction)
- Avg. max temp: 66.22C -> 51.32C (-22.5% reduction)
- Max temp: 77C -> 63C (-18.2% reduction)
- Energy: 1284.08J -> 1000.12J (-22.1% reduction)

Conclusion:

The on-demand governor ends up being strictly better for the bodytrack system while for the blackscholes, it sacrifices on runtime to save on energy, power, and a decrease in max temperature.