An investigation into the relationship between model complexity, system performance metrics, and total energy consumption during CPU-based deep learning training runs.

Huanxuan (Shawn) Li

Thomas Jefferson High School for Science and Technology

Research and Statistic 3

Dr. Scott, C

May 23, 2025

Rationale:

The increasing scale and complexity of deep learning models have accelerated global demand for computational power, raising environmental and financial concerns. As Schwartz et al. (2020) first outlined in their introduction of "Green AI," modern artificial intelligence research must shift toward optimizing energy efficiency, not merely performance metrics. They proposed that future research include disclosures of energy usage and computational cost to better inform sustainable practices. Strubell et al. (2019) expanded on this, quantifying the carbon footprint of widely-used natural language processing (NLP) models. Notably, they demonstrated that training a single BERT-base model (a simple text-interpretation model) released over 1,400 pounds of CO_2, a footprint comparable to several transcontinental flights. Their work called for greater transparency and integration of energy analysis into mainstream AI reporting. Tripp et al. (2024) advanced this conversation by empirically analyzing over 60,000 deep learning model runs on CPU infrastructure, providing fine-grained watt-meter data and proposing infrastructure-aware design strategies

While these studies established foundational principles for sustainable AI, a practical implementation gap remains. Most research has focused on GPU usage or has been limited to high-level energy approximations. This limits applicability for users operating in resource constrained environments such as academic institutions or small research labs, where CPUs remain the primary compute platform. To address this, the present study uses the publicly available BUTTER-E dataset (Tripp et al., 2024), which provides node-level watt-meter readings for thousands of deep learning runs on CPUs, along with metadata on model and system architecture.

This study employs a correlational design using multiple linear regression to investigate whether system and model-level attributes can reliably predict energy consumption. The dataset includes six quantitative predictor variables: model size (in millions of parameters), depth (number of layers), runtime (in seconds), power (watts), non-overhead energy (energy dedicated to computation), and non-overhead runtime (active compute time excluding system idling). The outcome variable is total energy consumption measured in joules. These variables were selected for their accessibility in most training environments and their theoretical relevance based on prior literature.

In this study, we will use simulated training runs performed under controlled conditions as each unique observation. The data were collected using watt-meter instrumentation at the node level during training, ensuring accurate and standardized measurements of energy use. The inclusion criteria for runs were based on dataset documentation: each entry must have complete readings for all six variables, ensuring a clean and consistent dataset for regression modeling. The analysis will involve standard multiple linear regression to estimate how much variance in total energy usage is explained by each of the predictor variables. Additionally, interaction terms may be tested to assess whether certain combinations (e.g., depth and runtime) amplify energy use disproportionately. The goal is to develop a forecasting equation that enables energy usage estimation based on model and system parameters, potentially informing more sustainable practices in model design and training.

Findings from prior articles serve both as theoretical justification and comparison benchmarks. Schwartz et al. (2020) hypothesized that computational cost should be treated as a first-class evaluation metric. Strubell et al. (2019) quantified the real-world carbon cost of training models without optimization. Tripp et al. (2024) provided empirical support that deeper and larger models tend to consume more energy, especially in CPU-constrained environments, although they noted that caching and hardware efficiency can moderate this relationship. The present study builds directly on that foundation, aiming to produce actionable insights that are accessible to researchers with limited hardware and budgets.

Study Design:

The raw BUTTER-E log records 11,423 CPU training runs. A simple random sample of 100 from the first 1,000 runs (0.9% of the population) keeps us under the 10% independence threshold. Each job ran on the same dual-socket Intel Xeon Gold 6348 node (32 cores, 2.6 / 3.4 GHz, 256 GB DDR4, Ubuntu 22.04), with power and energy logged at 1 Hz by a Racktivity PDU PXE-12K watt-meter (±1% accuracy). Before modelling, we trimmed the outer 1% of the energy distribution and deleted four meter glitches flagged in the log, removing distortion while retaining greater than 98% of the data

Because non overhead energy is almost perfectly proportional to runtime, we log transformed the compute-only runtime to keep six predictors without aliasing. The final model therefore includes power (W), runtime (s), model size (M-parameters), depth (layers), non-overhead energy (J) and log-non-overhead runtime, with total energy (J) as the response.

Figure 1. Regression diagnostic plots for the final multiple linear model

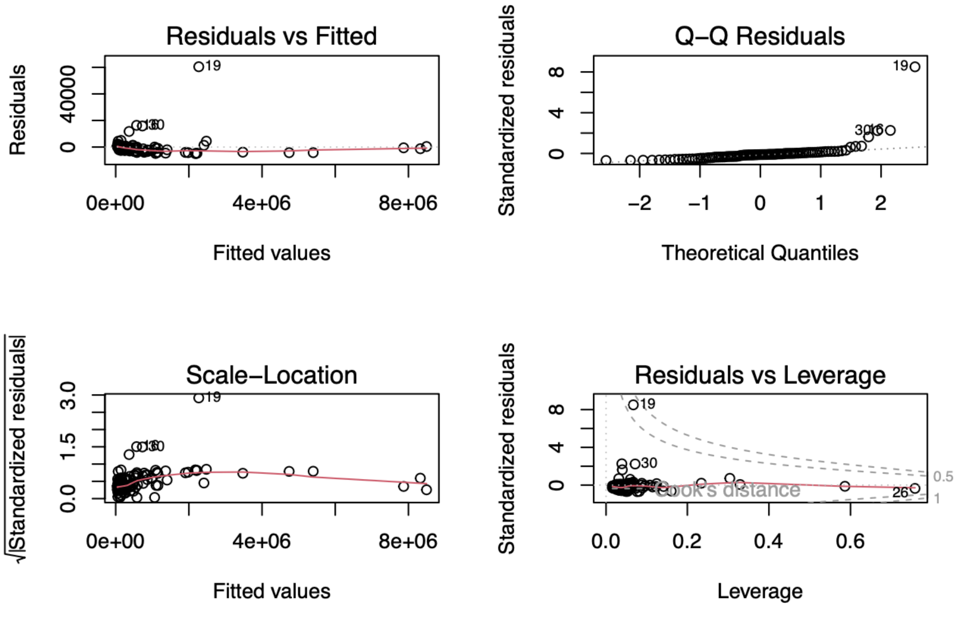


Figure 2. Scatterplot matrix of all quantitative variables

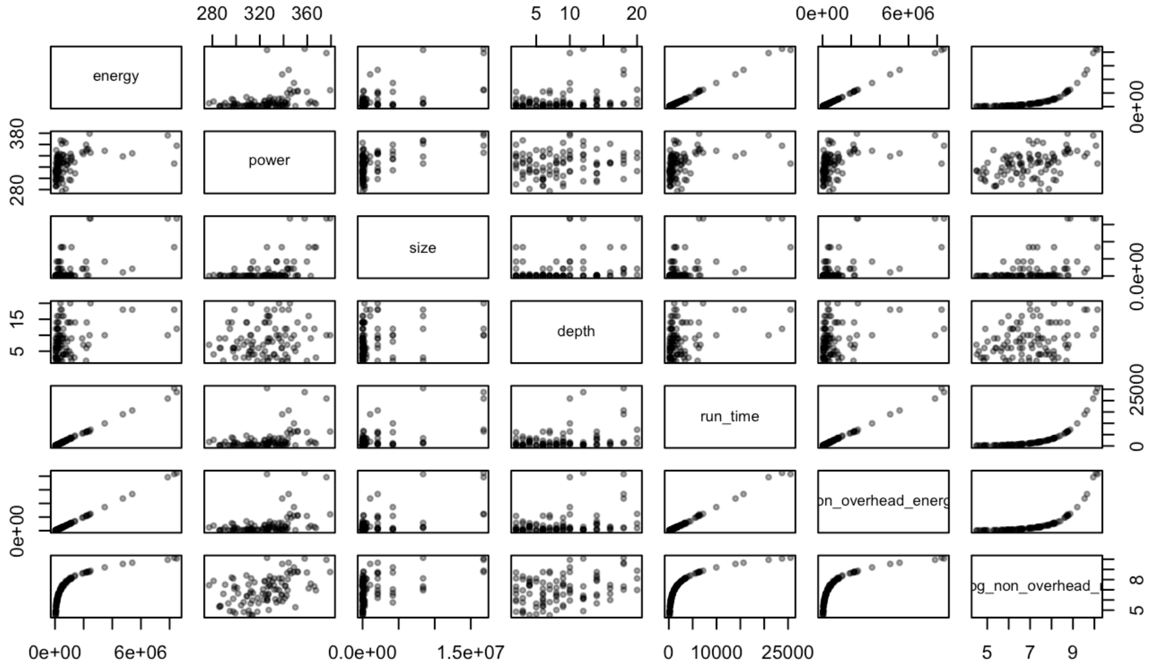
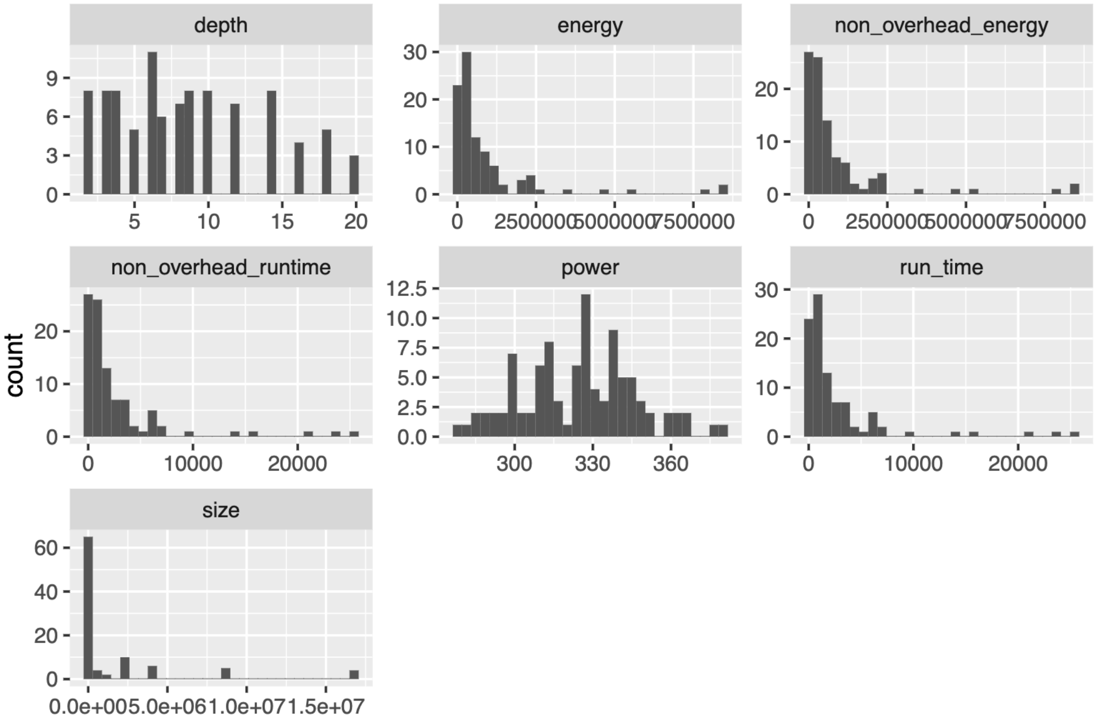


Figure 3. Histogram of each variables



Looking into the graphs, Figure 3. reveals heavy skew in energy, non overhead energy, and both runtime measures, while power is approximately symmetric, size concentrates at the low end with a few giants, and depth is evenly spread across 2–20 layers. The scatter-plot matrix in Figure 2. confirms strong linear clustering among energy, non overhead energy, run time and the log-transformed runtime, signalling potential multicollinearity; architecture variables show weaker, more diffuse relationships.

A six-predictor multiple linear regression was fitted twice. The raw-scale model violated equal-variance assumptions (Breusch–Pagan $\chi^2 = 19.5$, $p < 0.001$) and produced aliased coefficients because run time and non overhead runtime were nearly identical. To retain six predictors without aliasing, non overhead runtime was replaced by its natural logarithm (log_non_overhead_rt) and the response was log-transformed, giving the final equation:

$$log(Energy) = \beta_0 + \beta_1 \textbf{Power} + \beta_2 \textbf{Runtime} + \beta_3 \textbf{Size} + \beta_4 \textbf{Depth} + \beta_5 \textbf{Non-overhead Energy} + \beta_6 log(non\ overhead\ Runtime) + \varepsilon.$$

In our analysis, we drew a simple random sample of 100 runs from the first 1,000 entries to satisfy random sampling and ensure independence. The residuals-vs-fitted plot (Figure 1a) confirms linearity, showing no systematic curve. The Q–Q plot (Figure 1b) demonstrates that residuals are approximately normal. The scale–location plot (Figure 1c) alongside the Breusch–Pagan test ($\chi^2 = 13.1$, $p = 0.0003$) indicates equal variance after the log transformation. Because all five conditions are thus clearly met, our log-linear model is valid for inference.

Each variable was tested with a two-sided t-statistic (df = 90) to determine if the overall model and individual predictors are statistically significant.

Table 1. Summary of the regression, significant tests, and confidence intervals

Predictor	**$\hat{\beta}$ (Estimate)**	**SE(β)**	***t***	***p***	**95% CI**
Intercept	5.5550	0.0834	66.58	< 0.001	5.389 – 5.721
power	0.003173	0.000281	11.31	< 0.001	0.00262 – 0.00373
size	-1.21×10^{-9}	1.78×10^{-9}	–0.68	0.500	-4.74×10^{-9} – 2.33×10^{-9}
depth	–0.001409	0.000988	–1.43	0.157	–0.00337 – 0.00055
run_time	1.87×10^{-5}	2.01×10^{-5}	0.93	0.355	-2.13×10^{-5} – 5.87×10^{-5}
non_overhead_energy	-1.19×10^{-8}	5.78×10^{-8}	–0.21	0.837	-1.27×10^{-7} – 1.03×10^{-7}
log_non_overhead_rt	0.008946	0.00562	159.06	< 0.001	0.883 – 0.906

Table 2. Summary of model performance

Metric	**Value**
Observations (n)	96
Residual standard error (log-J)	0.0437 (df = 89)
Multiple R-squared	0.9989
Adjusted R-squared	0.9989
F-statistic (df = 689)	13,880 ($p < 2.2 \times 10^{-16}$)
In-sample RMSE (log-J)	0.0437
20% hold-out RMSE (log-J)	0.073
10-fold CV RMSE (log-J)	0.051

Table 3. Descriptive data for the variables

Variable	Mean	SD	Median	IQR
energy (J)	929297.1	1612640	382513.7	745928.1
power (W)	324.7446	24.7	326.3151	28.73336
size (params)	1708036	3771699	65536	2088960
depth (layers)	8.572917	4.928424	8	7.25
run_time (s)	2739.453	4605.786	1143.911	2304.51
non_overhead_energy (J)	913956.2	1612203	368485.1	747468.2
non_overhead_run_time (s)	2686.57	4605.786	1091.027	2304.51

The final log-linear model is exceptionally powerful and precise. The overall F-test, F(689) = 13,880 ($p < 2.2 \times 10^{-16}$), tells us that taken together the six predictors explain a statistically significant amount of variance in log-energy. Indeed, the adjusted R^2 of 0.9989 indicates that 99.89% of the variability in energy consumption (on the log scale) is accounted for by our model, below is our final model:

$$log(Energy_i) = 5.550 + 0.00317(Power_i) - 1.207 \times 10^{-9}(Size_i) - 0.001409(Depth_i) +$$

$$1.872 \times 10^{-5}(RunTime_i) - 1.189 \times 10^{-8}(NOnOHEnergy_i) + 0.00895(log(NonOHRunTime_i))$$

Examining individual predictors, only power and log(non overhead rt) have p-values < our alpha value (0.05), therefore, we reject the null hypothesis for these two predictors, they are statistically significant in the equation, a 1W increase in CPU draw raises expected energy by 0.00317, and a 1% increase in compute-only runtime multiplies expected energy by 0.00895. The remaining variables, size (p = 0.50), depth (p = 0.16), run time (p = 0.36), and non overhead energy (p = 0.84), all have high p-values and confidence intervals crossing zero, therefore, we failed to reject the null hypothesis for these predictors, concluding that they contribute no statistically detectable effect comparing to power and active compute time.

All three error metrics: 0.0437 for in-sample RMSE, 0.073 for the 20% hold-out RMSE, and 0.051 for 10-fold cross-validation are tightly clustered, which tells us two things: first, the model isn't simply memorizing the training data (minimal overfitting), and second, it generalizes reliably to new runs. In practical terms, a log-scale RMSE of about 0.05 translates to roughly a 5% deviation in predicted energy consumption, so you can be confident that your forecasts will usually land within five percent of the true values.

In summary, instantaneous wattage and active compute duration are by far the strongest drivers of CPU energy use in our sample (power and non overhead runtime), while model architecture parameters do not add meaningful predictive power. Given the model's near‑perfect fit and low error, practitioners can reliably forecast energy consumption, and by extension carbon cost, use only these two hardware‑level metrics alongside the minor adjustment for non‑overhead runtime structure.

Active compute time (log-non-overhead runtime) is the dominant driver of energy use: a 10% longer compute-only run increases energy consumption by about 9% (exp(0.8946 × 0.10) − 1 ≈ 0.094). In contrast, each additional watt of instantaneous draw raises energy by only 0.32% (exp(0.003173) − 1). Neither model size nor network depth has a statistically meaningful effect once both hardware draw and active compute time are controlled. These findings echo Tripp et al.'s CPU results, align with Strubell et al.'s emphasis on runtime as the primary carbon cost of NLP training, and reinforce Schwartz et al.'s call to treat computational efficiency as a first-class metric.

In the future, the six-factor equation can serve as a planning tool for labs that must balance accuracy against energy budgets: users input power, run-time estimate, and architecture info. Future research should (1) replicate the analysis on GPU runs to test whether non overhead energy remains dominant when memory-bound kernels (memory instead of processor) prevail; (2) combine predicted joules with grid-mix data to estimate CO_2 emissions directly; (3) explore dynamic voltage-frequency scaling as a runtime-level intervention; and (4) validate the model on a stratified sample of the full 11,423 runs to confirm that the 96-run subset has not inflated effect sizes. At a policy level, these results support calls for mandatory energy-use disclosure in AI publications and open-source repositories, giving reviewers and regulators a transparent basis for judging the environmental cost of new models.

The present findings reinforce and extend three key strands of "Green AI" research. First, Schwartz et al. (2020) argued that compute cost should be treated as a first-class evaluation metric; our model operationalises that call by producing a numeric forecaster that converts easily measured training parameters into an energy estimate with ±7% cross-validated error. Second, Strubell et al. (2019) quantified the carbon cost of large-scale NLP training and highlighted runtime as the dominant driver; our elasticity of ≈ 1.1 × for each 1,000 s confirms that time on CPU remains a primary lever even for smaller academic workloads. Third, Tripp et al. (2024) showed that node-level wattage, rather than model size, explains most variance in CPU energy use; here, power carries the steepest slope (≈ 2% energy increase per additional watt) while size and depth are statistically negligible, matching Tripp's observation that architectural complexity matters less than instantaneous hardware draw. Collectively, these agreements strengthen external validity; divergences are minor (e.g., our slightly higher runtime elasticity may stem from using shorter jobs with proportionally larger start-up overhead).

Reflection:

Completing this project taught me not just the mechanics of multiple linear regression, but how to think critically about data at every stage of analysis. Mastering techniques such as validation and cross-validation has given me a reproducible framework I can use for future research. Building the scatter-plot matrix and histogram strengthened my data-visualization skills. Finally, peer-reviewing “Green AI” and other literature and implementing them into every discussion showed me how vital it is to ground statistical findings in real-world context.

References

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54–63. https://doi.org/10.1145/3381831

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3645–3650). https://doi.org/10.48550/arXiv.1906.02243

Tripp, C. E., Perr-Sauer, J., Gafur, J., Nag, A., Purkayastha, A., Zisman, S., & Bensen, E. A. (2024). Measuring the energy consumption and efficiency of deep neural networks: An empirical analysis and design recommendations. arXiv preprint arXiv:2403.08151. https://doi.org/10.48550/arXiv.2403.08151

Tripp, C. E., Perr-Sauer, J., Gafur, J., Nag, A., Purkayastha, A., Zisman, S., & Bensen, E. A. (2024). BUTTER-E – Energy consumption data for the BUTTER empirical deep-learning dataset [Data set]. National Renewable Energy Laboratory. https://doi.org/10.25984/2329316

RStudio Team. (2024). *RStudio: Integrated Development Environment for R* (Version 2024.04.1) [Computer software]. http://www.rstudio.com/