Benchmarking Time Series Foundation Models on their Accuracy and Energy Consumption

Loïc Guibert [*1] **Benjamin Pasquier** [*1] **Frédéric Montet** [1] **Beat Wolf** [1]

Abstract

Our study presents a benchmark of ten time-series foundation models to quantify their accuracy–energy trade-off in zero-shot forecasting. Using an in-house and a public dataset (School and MeteoSwiss; univariate and multivariate variants), a fixed sliding-window protocol (context 512, horizon 64), and dual energy instrumentation (external PDU and CodeCarbon), we report sMAPE and NMAE accuracy metrics alongside runtime, energy (Wh), and Energy per Billion Parameters. Results show pronounced dataset dependence in accuracy, while efficiency is primarily architecture-driven: Chronos-Bolt achieves consistently low energy and latency, TimesFM attains the best MeteoSwiss accuracy at low energy cost, and Moirai-MoE exhibits substantially higher energy expenditure for comparable errors. This work informs decision-makers, developers, and end-users about the energy requirements of time-series foundation models and highlights the importance of considering energy alongside accuracy when evaluating models for adoption, while encouraging the systematic reporting of accuracy–energy trade-offs.

1. Introduction

Energy consumption has become a pivotal non-functional attribute of modern software systems. Regardless of its execution environment, the energy consumed by an application not only dictates components usage and operational costs, but also impacts the environment. Consequently, measuring the energy usage of software has emerged as a cornerstone of sustainability research, allowing for optimisations to be applied after the collection of such measurements. More specifically, the need for measurement has gained attention as the use of generative Artificial Intelligence (AI) increases; even when environmental impacts are not considered by the decision-makers, the operational costs of serving chatbots or other large AI models are directly linked to their energy consumption (Samsi et al., 2023; International Energy Agency, 2025; Alzoubi & Mishra, 2024).

Software and AI workflows use diverse computational patterns and resources (specialised accelerators, high data parallelism, memory intensity, communication-bound operations), bringing challenges to track their consumption (Tschand et al., 2025). Regarding AI workloads, the latter do not consume the same amount of energy due to varying characteristics, such as token length, batch size, hardware configuration, GPU resource sharing strategies, vocabulary sizes, and more (Samsi et al., 2023; Argerich & Patiño-Martínez, 2024).

Measuring AI energy usage presents significant challenges. Indeed, requests can be distributed in the cloud; modern infrastructure handles parallelised processes and multi-threading, while underlying hardware is abstracted through virtualisation. To quantify energy usage, three methods have been identified according to Fischer et al. (Fischer, 2025): (1) *static approaches*, assuming constant power draw; (2) *dynamic approaches*, tracking compute utilisation through code; (3) *external approaches*, taking measurements at the power socket (Bouza et al., 2023). In addition, a fourth method is the *intra-node approach*, which is more precise but necessitates specific equipment, such as a Baseboard Management Controller (BMC).

In dynamic approaches, software tools are explored by the research community due to their real-time capabilities and ease of integration. However, their measurements lack granularity due to restricted access to certain on-board components or hardware subsystems. One of the most well-known tools is CodeCarbon[1], which is widely recognised as the best tool to integrate into AI processes (Bouza et al., 2023; Jay et al., 2023; Ludvigsen, 2025). Other alternatives are

[*]Equal contribution [1]iCoSys, HEIA-FR, HES-SO University of Applied Sciences and Arts Western Switzerland, Fribourg, Switzerland. Correspondence to: Loïc Guibert <loic.guibert@hefr.ch>, Benjamin Pasquier <benjamin.pasquier@hefr.ch>, Frédéric Montet <frédéric.montet@hefr.ch>, Beat Wolf <beat.wolf@hefr.ch>.

Proceedings of the Swiss AI Days 2026, Martigny, VS & Fribourg, FR, Switzerland PMLR 309, 2026.

[1]https://github.com/mlco2/codecarbon

CarbonTracker (Anthony et al., 2020), Zeus (Chung et al., 2025a), MLPerf power (Tschand et al., 2025), or Ecologits (Rincé & Banse, 2025) or AI Energy Benchmarks (NeuralWatt, 2025).

Alternate software tools exist for system-wide measurements, also called software power meters: the most well-known projects are Scaphandre[2] and PowerAPI (Fieni et al., 2024), as well as the Perf Linux utility (Jay et al., 2023), all proposing different features. However, even if they can measure specific processes and/or components, such tools are less mature and precise than physical power meters. Besides the most interesting energy assessment tools already presented, a large variety of them have been released during the last years (Boavizta, 2025).

Large Language Model (LLM) leaderboards focusing on energy usage are receiving increasing interest, ranking models solely on their energy efficiency. The AI Energy score covers multiple tasks on three different GPUs capacities (Luccioni et al., 2025), while ML.ENERGY provides more details on tested LLMs but with a more restricted model selection (Chung et al., 2025b;a). However, such leaderboards for time series foundation models are not yet available. Therefore, the objective of our study is to propose an initial milestone to rank such models while taking their energy consumption into account. Consequently, we formulate two Research Questions (RQs): (RQ1) What is the energy consumption of time-series forecasting models?; and (RQ2) What is the trade-off between energy consumption and predictive performance?

Answering those questions would help practitioners be more mindful when it comes to using foundation models. Indeed, since such models show competitive performance compared to many traditional and deep learning methods, their usage could become more popular. Additionally, energy consumption is a concern for many; therefore, ranking those models with non-functional attributes, such as their energy impact, is meaningful.

The following Sections of this article follow the traditional scientific method. First, we present the method in Section 2, which details the datasets, models, experiments, and metrics. Then, results are presented in Section 3, discussed in Section 4 and concluded in Section 5, where limitations, ethics and further improvements are proposed.

2. Method

The goal of this work is to assess the energy consumption of modern and well-established foundation models for time series forecasting in relation to their predictive accuracy. The following Subsections present a benchmarking methodology designed to ensure fair and reproducible comparisons. We first describe the datasets and models considered, then introduce the experimental framework used to obtain consistent energy measurements, and finally detail the evaluation metrics employed in this study.

[2] https://github.com/hubblo-org/scaphandre

Table 1. Characteristics of the School and MeteoSwiss datasets and their variants.

	Multivariate	**Univariate**
School		
# Input Features	8	8 / 1
# Target features	8 (all buildings)	1 (single building)
Resolution	1 h	1 h
# Time steps	11,664	11,664
MeteoSwiss		
# Features	8	8 / 1
# Target features	8 (all variables)	1 (temperature)
Resolution	10 min	10 min
# Time steps	105,264	105,264

2.1. Datasets

In line with the study conducted in (Montet et al., 2025), we focus on one in-house dataset and one public dataset, referred to as School and MeteoSwiss respectively.

The School dataset reports hourly electricity consumption from eight buildings at our engineering school in Fribourg, Switzerland. The MeteoSwiss dataset provides meteorological observations from the Fribourg / Grangeneuve location at a 10-minute resolution, including variables such as atmospheric pressure, wind speed, and humidity (Federal Office of Meteorology and Climatology, 2026).

Table 1 summarises the main characteristics of these datasets and their two variants: the original multivariate version and a derived univariate version. Since some of the models considered in this study (see Subsection 2.2) are restricted to univariate forecasting and therefore process each component independently, these dataset variants enable a fairer comparison across models.

In contrast to our previous work, we restrict this study to only two datasets. This choice is motivated by our primary objective, which is to analyse and compare the energy consumption of different forecasting models rather than to evaluate predictive performance across a large collection of datasets.

Table 2. Overview of the models compared in the benchmark.

Model	Pred. Type	# Param.
Chronos Tiny (Ansari et al., 2024)	Uni.	8M
Chronos Large (Ansari et al., 2024)	Uni.	710M
Chronos Bolt Tiny (Ansari et al., 2024)	Uni.	9M
Chronos Bolt Base (Ansari et al., 2024)	Uni.	205M
Chronos 2 (Ansari et al., 2025)	Multi.	120M
Moirai Small (Woo et al., 2024)	Multi.	14M
Moirai Large (Woo et al., 2024)	Multi.	311M
Moirai-MoE Small (Liu et al., 2024)	Multi.	117M
Moirai-MoE Base (Liu et al., 2024)	Multi.	935M
TimesFM 2.0 (Das et al., 2024)	Uni.	500M

2.2. Foundation Models

Table 2 presents foundation models we consider in our benchmark. The ten selected models exhibit diverse characteristics: some support multivariate data (Multi.), while others are designed for univariate series (Uni.). Furthermore, model complexity varies significantly, with parameter counts spanning two orders of magnitude.

2.3. Experimental Framework

To conduct our measurements and benchmark, we rely on our in-house library FoMoTSF (Pasquier et al., 2026), which we extended to support on-the-fly energy consumption measurements. This library enables a fair and reproducible evaluation benchmark across different models, datasets, and configurations.

2.3.1. Energy Consumption Measurement

Two energy measurement approaches, external and dynamic, have been implemented into our experimental framework. This selection allows the obtained measurements to be cross-validated by comparing their values while observing their variations due to their different considered scopes. Both approaches are wrapped around the models benchmarking phase only, excluding preparation and processing steps.

The external approach relies on a Power Distribution Unit (PDU) with network connectivity, allowing real-time power data to be synchronously collected by a daemon probing the machine power drainage. The dynamic approach integrates the CodeCarbon library, configured in its *machine* tracking mode, to approximate total system energy consumption. This setup enables a direct comparison with the external measurement approach.

Both approaches collect the energy consumed for the benchmark, granularly to each configuration (model, dataset), at a fixed rate of 0.5 Hz, to avoid reading saturation of the PDU. This integration provides three metrics related to energy consumption, which are: (1) the Wh measured by CodeCarbon, (2) the Wh measured on the PDU during the evaluation process, and (3) the overall evaluation time.

2.3.2. Architecture

Experiments were conducted on a server-configured workstation powered by an NVIDIA RTX 6000 Blackwell Max-Q GPU (96 GB VRAM, 300 W power limit), an Intel Core i7-9700K CPU, and 32 GB of RAM. No cooling is provided except for built-in fans, and the power supply is connected directly to a datacenter-grade PDU[3]. The system runs on Ubuntu 24.04.3 LTS.

2.3.3. Evaluation Protocol

For evaluation, we adopt a sliding-window sampling strategy with a fixed stride of 16 to generate multiple input–target pairs from each time series dataset. The chosen stride value allows us to obtain a sufficient amount of pairs for both datasets. These samples are then fed to the foundation models to perform zero-shot forecasting. Figure 1 illustrates this sampling procedure.

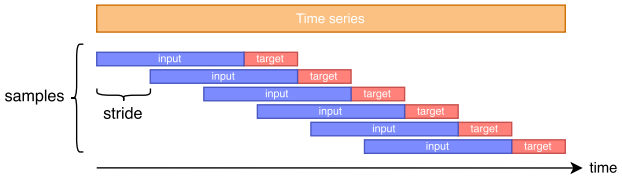

Figure 1. Sliding window sampling on the time series dataset. Each sample includes an input (context) and a target (ground truth for the prediction horizon).

Throughout all experiments, we systematically use an input length of 512 time steps and require the models to predict a forecasting horizon of 64 time steps. Models that support only univariate forecasting use the target variable alone as input when applied to univariate datasets, whereas multivariate-capable models systematically leverage all available variables to generate their predictions.

Regarding batch size during evaluation, we select the largest value that fits within the available GPU memory in order to maximise VRAM utilisation. However, given the 96 GB of VRAM available on our machine, some smaller models cannot fully saturate the GPU even at the maximum feasible batch size. Consequently, the GPU may not be fully utilised in these experiments.

[3] https://www.rnx.ch/e3meter/smart

2.4. Evaluation Metrics

In order to report model accuracy on both uni- and multivariate time series, we use scale-independent metrics such as sMAPE and NMAE instead of traditional metrics like MSE or MAE.

sMAPE (Symmetric Mean Absolute Percentage Error) evaluates the relative error as a percentage, symmetrically penalising over- and under-predictions. **NMAE** (Normalised Mean Absolute Error) expresses the total absolute error relative to the total magnitude of the true values.

For multivariate forecasting, metrics are computed per component and averaged to yield a single aggregated performance score across all target variables.

We also report the execution time (in seconds) and the total energy consumption (in Wh) for each model and dataset pair. To relate the energy cost of a model to its architectural complexity, we introduce the **EBP** (Energy per Billion Parameters) metric, which normalises the measured energy consumption E by the number of trainable model parameters P. EBP is computed as shown in Equation (1).

$$\textbf{EBP} = E \times \frac{10^9}{P} \tag{1}$$

Together, these metrics capture predictive accuracy, computational efficiency, and energy efficiency, enabling a comprehensive comparison of forecasting models under practical constraints.

3. Results

This Section presents the results of the experiments as described above. We first compare the forecasting accuracy of the foundation models across the four datasets. We then report resource usage in terms of inference time, energy consumption, and the EBP metric. Finally, we analyse trade-offs between forecasting accuracy and energy consumption, inference time, as well as differences between CodeCarbon and PDU measurements on the MeteoSwiss-Uni dataset.

3.1. Forecasting Accuracy

Table 3 reports the forecasting errors of all foundation models across all datasets, measured using sMAPE and NMAE metrics. Chronos-Bolt Base and Chronos Large achieve the best performance on the School dataset, while TimesFM exhibits the lowest errors on the MeteoSwiss datasets. In contrast, the smaller variants of Moirai, Moirai-MoE (Mixture of Experts) and Chronos exhibit the weakest performance, consistently ranking last across datasets.

Overall, the newer Chronos model family, such as Chronos Large and Chronos Bolt Base, consistently ranks among the top performers, often appearing as the best or second-best option. These results further highlight the influence of dataset characteristics on model performance, indicating that no single model really dominates across all datasets.

3.2. Resource Usage

Table 4 summarises the results obtained in terms of resource usage. The total time taken for the entire benchmark is indicated, alongside the energy it consumed and the EBP metric. The Chronos Bolt models in both sizes often outperform the others: they are among the fastest models while being fairly energy-efficient. Non-MoE Moirai models also present significant efficiencies. In contrast to the accuracy performances shown in Table 3, dataset characteristics do not influence results, with similarities found across datasets regardless of their size or dimensionality.

The total evaluation time varies significantly, both across models and across datasets. Some models take much more time compared to others, especially the Moirai ones. The Moirai-MoE models demonstrate the highest latency, certainly due to their specific architecture and the way their implementation handles inference. As for datasets, the differences between univariate and multivariate versions depend on the model, but this gap turns out to be negligible if Moirai or Chronos 2 models are used. However, the total evaluation time does not linearly fit the dataset sizes; despite the MeteoSwiss dataset being 10 times bigger than the School one, its total evaluation time is rather around 9 to 54 times longer.

The amount of energy used is sometimes correlated with the model sizes, but not systematically: indeed, comparable models such as Chronos Large & Moirai-MoE Base or Chronos Tiny & Chronos Bolt Tiny do not consume the same amount of energy, with the latter reflecting a gap of one order of magnitude on the MeteoSwiss multivariate dataset. This gap is highlighted by the EBP metric, which shows a high variability of values. As for the model flavours, such as small and large variations, the amount of energy significantly raise with larger variations, except for the Chronos Bolt family.

3.3. Accuracy–Energy Trade-off

Figure 2 illustrates the trade-off between the energy consumption measured by the PDU (Wh) and the forecasting error (NMAE). The scatter plot shows that most foundation models form a cluster around an NMAE of ~ 0.65 with an energy consumption below 5 Wh, indicating a favourable balance between accuracy and efficiency. TimesFM stands out with the lowest NMAE while maintaining a similarly low energy footprint. In contrast, although Moirai-MoE Base achieves the second-best forecasting accuracy, its evaluation incurs a substantially higher energy cost, reaching

Table 3. Forecasting performance of foundation models on a prediction horizon of 64 time steps, evaluated in a zero-shot setting. The best score per row is in bold; the second best is underlined.

		Moirai		Moirai-MoE		Chronos		Chronos Bolt		CH2†	TFM‡
	Metrics	**Small**	**Large**	**Small**	**Base**	**Tiny**	**Large**	**Tiny**	**Base**		
School-Uni	*sMAPE*	28.87	24.36	27.99	24.27	24.46	**21.52**	23.31	<u>22.14</u>	45.56	23.87
	NMAE	0.342	0.283	0.340	0.287	0.298	<u>0.261</u>	0.269	**0.260**	0.728	0.274
School-Multi	*sMAPE*	23.40	20.46	22.65	20.25	21.85	<u>19.90</u>	20.29	**19.40**	33.14	21.38
	NMAE	0.269	0.235	0.263	0.233	0.257	0.232	<u>0.230</u>	**0.223**	0.458	0.248
MeteoSwiss-Uni	*sMAPE*	42.27	41.67	41.60	39.75	44.22	41.81	40.94	<u>39.36</u>	41.12	**32.18**
	NMAE	0.658	0.645	0.622	<u>0.580</u>	0.716	0.60	0.663	0.646	0.642	**0.521**
MeteoSwiss-Multi	*sMAPE*	66.53	64.07	63.17	63.29	65.44	<u>61.16</u>	64.72	62.16	62.14	**59.04**
	NMAE	0.759	0.845	1.230	0.870	0.892	0.88	0.860	0.933	<u>0.683</u>	**0.627**

† CH2 = Chronos 2, ‡ TFM = TimesFM 2.0.

$704\ Wh$. Its smaller variant may be a good alternative, as it consumes roughly five times less energy while still achieving the third-best NMAE. Overall, these results confirm that smaller models generally exhibit lower energy consumption, highlighting a clear accuracy–efficiency trade-off.

3.4. CodeCarbon vs. PDU Energy Measurement

The two energy measurement approaches we used are contrasted in Figure 3. The model rankings of both approaches are strongly correlated, confirming that software-based tools can effectively capture relative energy differences. The PDU consistently reports slightly higher energy consumption than CodeCarbon across nearly all models: this gap is expected, as the PDU captures the complete machine power drainage, including power loss from the power supply unit, the cooling and auxiliary components that software-level tools have no access to. Except for TimesFM 2.0, the gap between the two approaches is around $6\ \%$.

However, measurements of the lightest models clearly show CodeCarbon values higher than the PDU ones, which is physically not possible. This situation shows the importance of performing measurements that can be verified.

3.5. Inference Time

While Subsection 3.2 reports the total evaluation time for each model, we focus here on single-sample inference time. Figure 4 shows the time required by each foundation model to generate a forecast for one sample, using a context length of 512 time steps and a forecasting horizon of 64 time steps. Although inference times generally follow the same trend as validation times, some models, such as Chronos-2, Moirai Large, and Moirai Small, exhibit among the longest validation times in Subsection 3.2 while remaining highly competitive in single-sample inference. Overall, as expected, smaller models tend to achieve lower inference latency.

4. Discussion

In this study, we address two main research questions: (RQ1) what is the energy consumption of foundation models for time series forecasting, and (RQ2) how does energy usage trade off against forecasting performance? To address these questions, we conducted an extensive benchmarking study of several foundation models evaluated on four in-house datasets, using a unified evaluation protocol and consistent metrics to ensure a fair and systematic comparison.

The difference in total evaluation time among dataset sizes can be explained by the fact that the batch size used for smaller models cannot be increased as high as the number of predictions to be performed, leading to less efficient inferences.

Regarding **forecasting accuracy**, we cannot identify a single foundation model that consistently outperforms all others, as the differences in error across models generally fall within a similar range. Nevertheless, Chronos-Bolt models, together with Chronos Large, tend to achieve the strongest overall performance, while TimesFM remains the top-performing model on the MeteoSwiss datasets. This observation is consistent with the conclusions of our previous study (Montet et al., 2025). We therefore recommend applying our methodology to specific use cases.

Regarding **resource usage**, our results show that the longest evaluation processes do not necessarily consume more energy. Since energy consumption is the product of power draw and operating time, these processes suggest that the machine is being underutilised. A tuning phase aiming for optimal occupation of the machine should be performed to obtain efficient provisioning. More specifically for GPUs,

Table 4. Forecasting resource usage of foundation models on a prediction horizon of 64 time steps, evaluated in a zero-shot setting. The best score per row is in bold; the second best is underlined. The total evaluation time (*time*) is in s, and the energy measured by the PDU (*energy*) is in Wh.

		Moirai		Moirai-MoE		Chronos		Chronos Bolt		CH2†	TFM‡
	Metrics	Small	Large	Small	Base	Tiny	Large	Tiny	Base		
School-Uni	*Time*	54.29	56.15	212.38	733.18	13.12	67.85	9.07	**8.64**	56.52	27.04
	Energy	**0.02**	0.25	17.18	74.34	0.28	6.38	0.06	0.07	0.13	1.32
	EBP	1.43	0.80	146.83	79.51	35.00	8.99	6.67	**0.34**	1.08	2.64
School-Multi	*Time*	59.88	56.14	213.30	733.61	35.87	484.77	13.18	**12.85**	55.89	163.37
	Energy	0.18	0.19	16.93	74.11	2.46	51.45	0.21	0.29	**0.13**	7.78
	EBP	12.86	**0.61**	144.70	79.26	307.50	72.46	23.33	1.41	1.08	15.56
MeteoSwiss-Uni	*Time*	3231.27	2995.12	4451.45	9483.94	81.54	613.64	**54.32**	56.03	3028.16	233.06
	Energy	0.73	1.67	162.79	626.82	3.32	62.12	**0.70**	0.75	1.30	9.92
	EBP	52.14	5.37	1391.37	670.40	415.00	87.49	77.78	**3.66**	10.83	19.84
MeteoSwiss-Multi	*Time*	3117.37	3193.61	4643.36	9500.64	283.70	4564.87	**89.13**	90.85	3118.82	1513.76
	Energy	**0.70**	1.66	163.21	703.41	23.62	492.20	2.04	2.30	0.87	69.07
	EBP	50.00	**5.34**	1394.96	752.31	2952.50	693.24	226.67	11.22	7.25	138.14

† CH2 = Chronos 2, ‡ TFM = TimesFM 2.0.

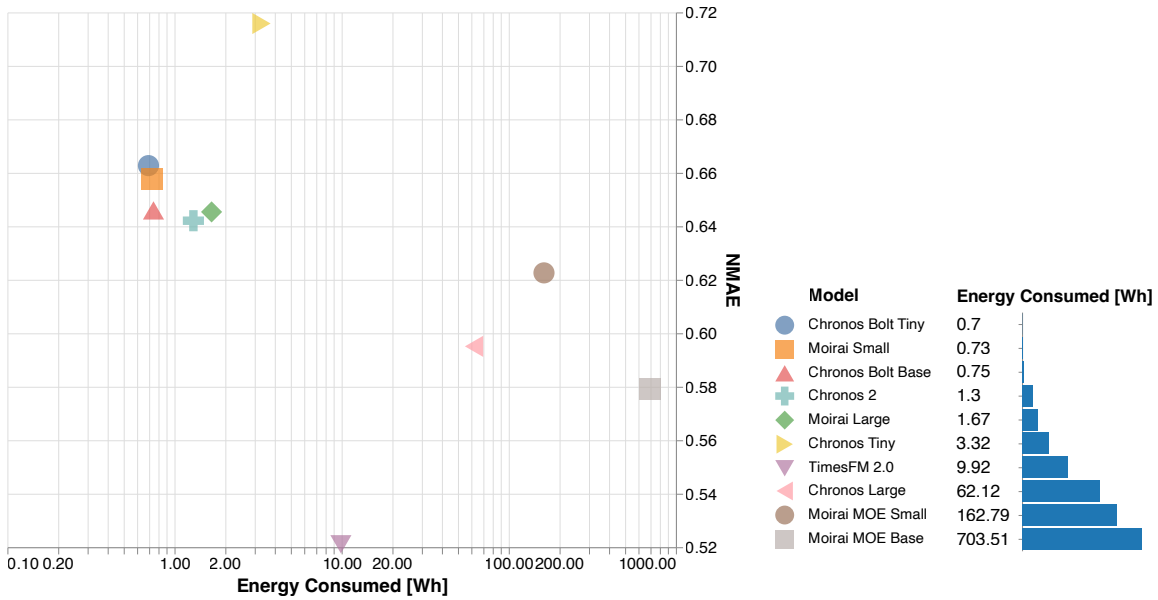

Figure 2. Comparison of the foundation models trade-off between energy consumption, measured with the PDU (log scale), and forecasting errors (NMAE) on the MeteoSwiss-Uni dataset.

the batch size parameter greatly influences their usage factor.

The EBP metric serves as a key indicator of architectural efficiency, showing a uncorrelated relationship between energy consumption and model size. This relation becomes evident when comparable models show vastly different energy profiles: some models are significantly better optimised regardless of their parameter count. In accordance with our contextual experience, we can state that Moirai-MoE small is the worst choice in terms of energy efficiency and is not even better in terms of forecast accuracy.

On the **energy measurement** approaches, software-based power meters typically rely on estimation models (RAPL for CPUs or NVML for GPUs). At very low machine utilisation, these estimations can become unreliable due to how they sample rest states or background system processes. This

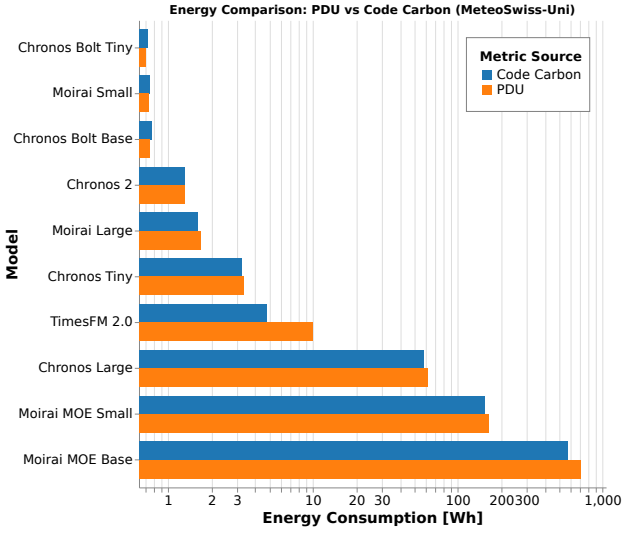

Figure 3. Comparison of PDU and CodeCarbon energy measurement on all model across the MeteoSwiss Univariate dataset. The X-axis has a logarithmic scale

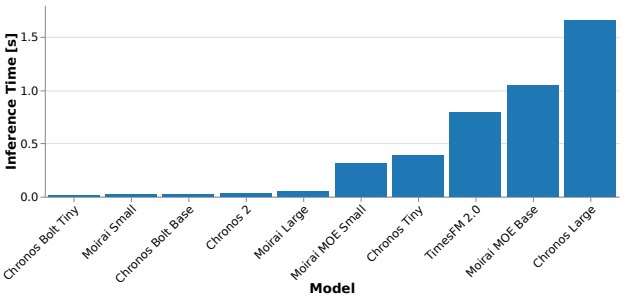

Figure 4. Computational times taken by each model to generate a forecast for one single sample of MeteoSwiss-Uni dataset.

finding highlights a critical limitation: while software tools are suited for ranking high-power models, they may lack the granularity required for lighter models. Users must therefore make sure that their measurement solution is suited to their context and must also ensure that the measurement period is long enough to collect a representative amount of measurements. Moreover, a high sampling frequency of the instant power drainage is required to capture transient power spikes, regardless of the used approach.

Our evaluation also reveals that while integrated software tools like CodeCarbon effectively capture relative energy differences for model comparisons, they should be used with caution for absolute measurements. Relying on these values as ground truths in production deployments may introduce imprecisions, potentially leading to under-evaluated environmental impacts.

Our results indicate that evaluation time does not always follow the same trend as single-sample inference time. This finding suggests that some models handle batched data more efficiently than others. Regarding the Chronos and Chronos-Bolt models, their longer evaluation times can be partially explained by additional preprocessing steps made by our library, a required step to properly format the input time series. However, the notably slow evaluation times observed for the Moirai models cannot be fully explained by our experimental setup, indicating that the official implementation provided by their providers is inefficient.

5. Conclusion

This study provides a comprehensive benchmark of ten time-series foundation models, establishing a critical baseline for evaluating the accuracy-energy trade-off in zero-shot forecasting. Our findings highlight that while forecasting accuracy is heavily dependent on the dataset, energy efficiency is primarily a product of model architecture. Besides, the introduced Energy per Billion Parameters (EBP) metric revealed an uncorrelated relationship between energy use and model size.

In terms of accuracy, no single model dominates all scenarios. Chronos-Bolt Base and Chronos Large achieved the best performance on the School datasets, while TimesFM 2.0 exhibited the lowest errors on the MeteoSwiss datasets. As for their efficiency profiles, Chronos-Bolt models consistently demonstrated low energy consumption and latency. Conversely, the Moirai-MoE Base model incurred a substantially higher energy cost—reaching 568 Wh on the MeteoSwiss-Uni dataset—for comparable errors, making it the least efficient choice in this benchmark.

While software tools like CodeCarbon effectively capture relative energy differences between models, they can provide unreliable measurements for the lighter models. Moreover, the evaluation process should be long enough to obtain a significant number of measurements to ensure a certain degree of accuracy. Physical PDU measurements remain the ground truth, as they capture complete machine drainage, including cooling and auxiliary components; although it is preferable that the machine is occupied at its maximal capacity.

Finally, if one wants to determine which models reach a better trade-off between forecast accuracy and energy usage, we encourage them to apply our methodology to obtain unbiased and meaningful information suited to the specificity of their use cases.

Acknowledgements

This work has been financed by the University of Applied Sciences and Arts Western Switzerland (HES-SO) as part of a young researcher grant to Beat Wolf, with the name "Foundation Models for Time Series Forecasting".

Limitations

The experiments were conducted on a single machine: consequently, the results may vary across different hardware configurations. Our measurement of the *whole machine* consumption via the PDU includes power loss from auxiliary components and power supply unit inefficiencies, which may not generalise to datacenter-scale environments with different cooling or power-sharing strategies.

To ensure consistency, we selected the largest batch size that fit within the available 96 GB of VRAM. However, smaller models often could not fully saturate the GPU even at maximum feasible batch sizes. This lack of full GPU utilisation means that the recorded energy consumption might not reflect the peak efficiency these models could achieve in a production environment.

The observed performance, particularly regarding execution time and energy expenditure, is heavily influenced by the models' official software implementations. For example, the Moirai models exhibited notably long evaluation times that cannot be fully explained by our experimental setup alone. These results may therefore reflect current implementation bottlenecks rather than the theoretical limits of the model architectures themselves.

Ethical Statement

Our model evaluation process has required substantial usage of compute time that also involves a GPU. With prototyping and failed runs, we estimate that all our experiments occupied our workstation for approximately 35 hours, with an average power drainage around 337 W. Our contribution has therefore released around *554 grams of* CO_2, considering a carbon intensity of 47 grams of CO_2 per KWh.

Aside from its environmental impact, this work does not raise any other ethical issues. The models and datasets used in this benchmark are either all publicly available or created by the authors. If any, all licence terms have been respected. We do not use any personal data in this work.

References

Alzoubi, Y. I. and Mishra, A. Green artificial intelligence initiatives: Potentials and challenges. *Journal of Cleaner Production*, 468:143090, 2024. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2024.143090. URL https://www.sciencedirect.com/science/article/pii/S0959652624025393.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda Arango, S., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Gordon Wilson, A., Bohlke-Schneider, M., and Wang, Y. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Ansari, A. F., Shchur, O., Küken, J., Auer, A., Han, B., Mercado, P., Rangapuram, S. S., Shen, H., Stella, L., Zhang, X., Goswami, M., Kapoor, S., Maddix, D. C., Guerron, P., Hu, T., Yin, J., Erickson, N., Desai, P. M., Wang, H., Rangwala, H., Karypis, G., Wang, Y., and Bohlke-Schneider, M. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. URL https://arxiv.org/abs/2510.15821.

Anthony, L. F. W., Kanding, B., and Selvan, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models, 2020. URL https://arxiv.org/abs/2007.03051.

Argerich, M. F. and Patiño-Martínez, M. Measuring and improving the energy efficiency of large language models inference. *IEEE Access*, 12:80194–80207, 2024. doi: 10.1109/ACCESS.2024.3409745.

Boavizta. NocoDB, 2025. URL https://db.boavizta.org/dashboard/#/base/e3ba7aca-a9a7-4984-ad8d-8949ba47e305/md_kvka0tcnh8dyrw.

Bouza, L., Bugeau, A., and Lannelongue, L. How to estimate carbon footprint when training deep learning models? a guide and review. *Environmental Research Communications*, 5(11):115014, nov 2023. doi: 10.1088/2515-7620/acf81b. URL https://doi.org/10.1088/2515-7620/acf81b.

Chung, J.-W., Ma, J. J., Wu, R., Liu, J., Kweon, O. J., Xia, Y., Wu, Z., and Chowdhury, M. The ml.energy benchmark: Toward automated inference energy measurement and optimization, 2025a. URL https://arxiv.org/abs/2505.06371.

Chung, J.-W., Ma, J. J., Wu, R., Liu, J., Kweon, O. J., Xia, Y., Wu, Z., and Chowdhury, M. The ML.ENERGY benchmark: Toward automated inference energy measurement and optimization. In *NeurIPS Datasets and Benchmarks*, 2025b.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Federal Office of Meteorology and Climatology. SwissMetNet - Fribourg / Grangeneuve , 1 2026. URL https://data.geo.admin.ch/browser/index.html#/collections/ch.meteoschweiz.ogd-smn/items/gra.

Fieni, G., Acero, D. R., Rust, P., and Rouvoy, R. PowerAPI: A Python framework for building software-defined power meters. *Journal of Open Source Software*, 9(98):6670, June 2024. doi: 10.21105/joss.06670. URL https://hal.science/hal-04601379.

Fischer, R. Ground-truthing ai energy consumption: Validating codecarbon against external measurements, 2025. URL https://arxiv.org/abs/2509.22092.

International Energy Agency. Energy and AI – Analysis - IEA, 4 2025. URL https://www.iea.org/reports/energy-and-ai.

Jay, M., Ostapenco, V., Lefevre, L., Trystram, D., Orgerie, A.-C., and Fichel, B. An experimental comparison of software-based power meters: focus on cpu and gpu. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pp. 106–118, 2023. doi: 10.1109/CCGrid57682.2023.00020.

Liu, X., Liu, J., Woo, G., Aksu, T., Liang, Y., Zimmermann, R., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Moirai-moe: Empowering time series foundation models with sparse mixture of experts, 2024. URL https://arxiv.org/abs/2410.10469.

Luccioni, S., Gamazaychikov, B., Strubell, E., Hooker, S., Jernite, Y., Mitchell, M., and Chamberlin, S. Ai energy score leaderboard - december 2025. https://huggingface.co/spaces/AIEnergyScore/Leaderboard, 2025.

Ludvigsen, K. G. A. How to estimate and reduce the carbon footprint of machine learning models, 1 2025. URL https://towardsdatascience.com/how-to-estimate-and-reduce-the-carbon-footprint-of-machine-learning-models-49f24510880/.

Montet, F., Pasquier, B., and Wolf, B. Benchmarking foundation models for time-series forecasting: Zero-shot, few-shot, and full-shot evaluations. *Computer Sciences & Mathematics Forum*, 11(1), 2025. ISSN 2813-0324. doi: 10.3390/cmsf2025011032. URL https://www.mdpi.com/2813-0324/11/1/32.

NeuralWatt. Ai energy benchmarks: A framework for measuring ai model energy consumption, 2025. URL https://github.com/neuralwatt/ai_energy_benchmarks.

Pasquier, B., Guibert, L., Montet, F., and Wolf, B. FoMoTSF - Foundation Models for Time Series Forecasting. GitLab repository, 2026. https://gitlab.forge.hefr.ch/fomotsf/fomotsf-public (accessed on 04.03.2026).

Rincé, S. and Banse, A. Ecologits: Evaluating the environmental impacts of generative ai. *Journal of Open Source Software*, 10(111):7471, 2025. doi: 10.21105/joss.07471. URL https://doi.org/10.21105/joss.07471.

Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., and Gadepally, V. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9, 2023. doi: 10.1109/HPEC58863.2023.10363447.

Tschand, A., Rajan, A. T. R., Idgunji, S., Ghosh, A., Holleman, J., Kiraly, C., Ambalkar, P., Borkar, R., Chukka, R., Cockrell, T., Curtis, O., Fursin, G., Hodak, M., Kassa, H., Lokhmotov, A., Miskovic, D., Pan, Y., Manmathan, M. P., Raymond, L., John, T. S., Suresh, A., Taubitz, R., Zhan, S., Wasson, S., Kanter, D., and Reddi, V. J. Mlperf power: Benchmarking the energy efficiency of machine learning systems from μwatts to mwatts for sustainable ai. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1201–1216, 2025. doi: 10.1109/HPCA61900.2025.00092.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/woo24a.html.