Domain Adaptation Under Wireless Network Constraints: When Does It Become Green?

Anonymous Authors[1]

Abstract

The deployment of data-driven models in 6G wireless networks is increasingly challenged by frequent distribution shifts that degrade performance over time. Unsupervised Domain Adaptation (UDA) offers an alternative approach by adapting the trained model to a shifted domain without requiring labels. However, UDA pipelines are often more complex than single-task training due to additional modules and optimization procedures, raising a practical question: do the benefits of adaptation come at a higher energy cost, and how does this trade-off compare to retraining when labeling effort is also considered? In this work, we investigate the energy consumption of UDA and compare it to single task. We further propose a way to determine the minimum number of target domains for which UDA becomes more energy-efficient than retraining, taking into account the labeling cost. Our results aim to clarify when UDA should be preferred over classical train-from-scratch approaches from an energy and labeling-aware perspective.

1. Introduction

The sixth generation (6G) of wireless communication is expected to have a tenfold increase in traffic load, enabling higher connectivity and better services (Ericsson, 2025). However, this growth is accompanied by extreme dynamism and heterogeneity across deployment scenarios: cell densities, traffic patterns, channel models, and user mobility vary significantly in different environments. As a result, data-driven models trained under a single set of conditions such as a specific urban micro-cell with a particular fading profile, consistently fails to generalize to other deployments, a phenomenon known as domain shift. Large generative models like foundation models can in principle bridge this generalization gap, but their high inference latency (often hundreds of milliseconds to hundreds of seconds) conflicts with the strict real-time requirement constraints of many wireless tasks, such as channel estimation, interference classification, and handover decisions, which demand sub-second to microsecond responses (Maatouk et al., 2024; Lin et al., 2022). Consequently, such large models are impractical for edge and base-station deployment.

Unsupervised Domain Adaptation (UDA) offers a promising alternative by enabling relatively small models, trained on abundant labeled data from a source domain (e.g., simulations or laboratory environments), to adapt to an unlabeled target domain (e.g., a live rural or dense-urban deployment) without requiring expensive ground-truth annotations in the field (Ismail Fawaz et al., 2025). For wireless applications where signal-to-noise ratio, multipath fading, and user mobility vary substantially, UDA can dramatically reduce the need for site-specific data collection and labeling, thereby accelerating the scalable deployment of AI-driven systems. Yet, wireless networks impose additional practical constraints beyond domain shift: models must be lightweight for continuous inference, amenable to rapid retraining on short timescales, and energy-efficient to support the sustainability goals that 6G explicitly commits to (Ericsson, 2023a; Next G Alliance Green G Working Group, 2025). These requirements challenge conventional assumptions about domain adaptation, which has largely been studied in computer vision and natural language processing where latency and energy budgets are far less restrictive.

In this work, we evaluate the adaptability of time-series classification tasks under realistic wireless conditions, with a particular focus on measuring algorithmic efficiency from a green AI perspective. Intuitively, avoiding full retraining from scratch and bypassing costly labeling sounds more efficient, but whether UDA actually delivers a favorable trade-off among energy consumption, labeling cost, adaptation speed, and generalization performance remains an open question, both theoretically and empirically. Our analysis aims to check where UDA outperforms both training from scratch (expensive in energy and labels) and fine-tuning large pre-trained models (too slow and power-hungry), providing practical guidelines for deploying sustainable,

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

real-time UDA in next-generation wireless systems.

2. Context and notations

2.1. Wireless Network Constraints

Wireless networks present a unique set of constraints that challenge conventional AI pipelines. First, wireless data is inherently **temporal** in nature, with correlations and non-stationarities that standard i.i.d. assumptions violate (O'shea & Hoydis, 2017; Sun et al., 2018). Second, **label scarcity** is a recurrent issue, as the ground-truth annotation for many use cases like channel states, interference sources, or traffic classes in live deployments is expensive and often impractical, on top of which it raises privacy aspects (Soares et al., 2023; Saffar et al., 2019). Third, **domain shifts** frequently arise from changes in the spatio-temporal environment (user mobility, weather, network load, geographical deployment, countries, or hardware configurations), occurring far more often than in typical computer vision or NLP domains. In particular, **covariate shift** is frequently encountered in wireless networks due to their dynamicity (Raza et al., 2014; Raghuram et al., 2021; Talak et al., 2018). Fourth, any practical solution must be **lightweight for inference and amenable to rapid or offline retraining** on short time scales (micro- or milliseconds) to comply with operational constraints, ruling out large foundation models or slow online learning. Finally, there is a need for **sustainable AI** (Ericsson, 2023a; Bothe et al., 2025; Next G Alliance Green G Working Group, 2025; Lin et al., 2022; Lai et al., 2026). Indeed, ICT sector used about 4% of the global electricity in the use stage and represented about 1.4% of the global GHG emissions in 2020 (Buzzi et al., 2016; Malmodin et al., 2024). It holds the potential to enable a 15% reduction in cross-industry emissions by 2030 through connectivity-driven solutions such as smart and autonomous networks, smart building management and connected electric vehicle charging infrastructure (Ericsson, 2023b). If we integrate AI to support these green networks, then AI itself must be sustainable in return. Models must therefore have low energy footprints during both training and inference.

Domain shift and label scarcity correspond to the main assumptions of UDA techniques, which make such approaches particularly indicated for wireless applications. However, the application of UDA to time series presents unique challenges due to the inherent characteristics of such a data type. Unlike images, time series data involves temporal ordering, trends, seasonality, and varying frequencies, all of which influence analysis and performance. These factors contribute to the increased complexity of UDA for time series. Additionally, To the best of our knowledge there are no previous work comparing the energy consumption of UDA with retraining from scratch standard classification approaches on new datasets. Within this realistic setting, we investigate the following central question: *When does Unsupervised Domain Adaptation (UDA) become "green" and offer a favorable compromise among energy efficiency, labeling cost, adaptation speed, and generalization performance?* Our contributions identify the regimes in which UDA surpasses both training from scratch which incurs high energy and labeling costs and fine-tuning large pre-trained models, which can be slow and computationally intensive. Based on these findings, we offer practical guidelines for applying UDA in next-generation wireless systems.

2.2. UDA for Time series classification

Let $\mathbf{x} = (x^1, \ldots, x^d) \in \mathbb{R}^{t \times d}$, be a time series where d is the number of features (or channels) and t its length. In classification, the goal is to predict $y \in \{0, \ldots, c-1\}$ from a labeled set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from distribution $\mathcal{D}$. In the telecom context, a representative example is traffic classification, where inputs are features such as packet inter-arrival times and outputs are traffic types, e.g., voice, data, SMS.

UDA aims to transfer knowledge from a labeled source domain to an unlabeled target domain by reducing the discrepancy (or also called shift) between their distributions, without access to target labels during training. This shift may occur at the level of the input distribution (covariate shift), the label distribution (label shift), or the relationship between inputs and labels (concept drift). This paper focuses on covariate shift, following the mathematical setting of (Ben-David et al., 2010). Consider two distributions over $\mathcal{X} \times \mathcal{Y}$: a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$. A UDA algorithm is provided with n_S i.i.d. a labeled source dataset $S = \{(\mathbf{x}_i^S, y_i^S)\}_{i=1}^{n_S}$ drawn i.i.d. from $\mathcal{D}_S$, and an unlabeled target dataset $T = \{\mathbf{x}_i^T\}_{i=1}^{n_T}$ drawn i.i.d. from $\mathcal{D}_T^X$, the marginal distribution of $\mathcal{D}_T$ over $\mathcal{X}$. The goal is to learn a classifier $\eta : \mathcal{X} \to \mathcal{Y}$ with low target risk:

$$R_{\mathcal{D}_T}(\eta) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}_T}\left[\eta(\mathbf{x}) \neq y\right],$$

without access to target labels. Under the covariate shift assumption, $p_S(\mathbf{x}) \neq p_T(\mathbf{x})$ while $p_S(y \mid \mathbf{x}) = p_T(y \mid \mathbf{x})$.

3. Energy Consumption UDA vs TSC

3.1. AI & Carbon footprint estimation

Quantifying the energy footprint of AI models is critical not only at training and inference time, but across the full AI lifecycle: data collection and preprocessing, model training, validation, inference, monitoring for distribution shift, and retraining upon model staleness. Energy consumption is typically measured in kWh via hardware power sampling (e.g., RAPL for CPUs, NVML for GPUs) or estimated from FLOPs and hardware TDP (Patterson et al., 2021). To relate energy to environmental impact, several open-source li-

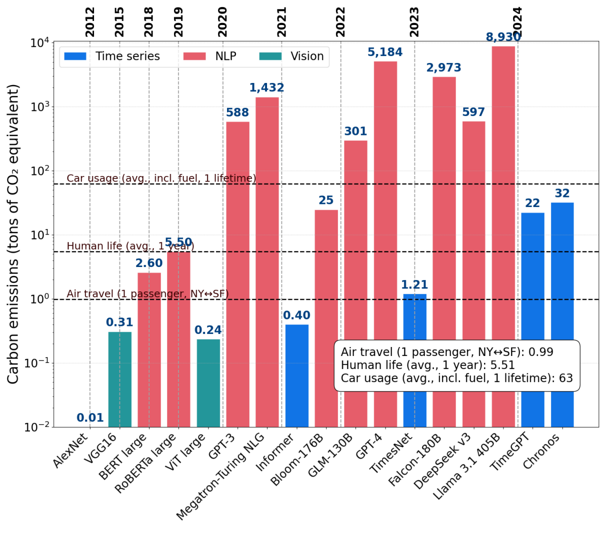


Figure 1. Estimated carbon emissions from training select AI models and real-life activities, 2012–24.

braries support end-to-end carbon accounting within Python-based workflows, including CarbonTracker (Anthony et al., 2020), Eco2AI (Budennyy et al., 2022), and CodeCarbon (Courty et al., 2024b). These tools estimate the carbon footprint CF (in gCO_2eq) as: $\text{CF} = \text{CI} \times \text{PUE} \times E$, where CI ($gCO_2eq$/kWh) is the carbon intensity of the local electricity grid, PUE ≥ 1 is the Power Usage Effectiveness of the compute infrastructure, and E (kWh) is the total energy consumed by the stage under consideration. This unified approach is very beneficial since it allows quantifying the different stages of the of the AI life cycle management (LCM-AI) on the same scale.

The scale of these costs varies dramatically across model families (see Fig. 1). Training a single BERT-base model emits approximately 2.6 tons od CO_2eq, while TimeGPT can reach 22 tons of CO_2eq (Strubell et al., 2019; Lacoste et al., 2019). For some families of models, such as large language models (LLM), the total life cycle cost is dominated by the inference phase, using much higher resources (65%) as compared to training (35%) due to the millions of daily queries (Desislavov et al., 2023; Wu et al., 2022). In contrast, lightweight models such as logistic regression or small CNNs consume several orders of magnitude less energy, making model selection a critical lever for Green AI (Mao et al., 2021). These observations motivate our focus on relatively small, efficient UDA models suited to the resource-constrained wireless deployments.

3.2. UDA versus Independent Classification: Cost Analysis and Break-Even Estimation

To quantify the efficiency gains of UDA when adapting a single source model to multiple target domains, we compare its computational and environmental cost against an independent classification baseline, in which a separate supervised model is trained from scratch for each target domain.

Pipeline Stages. Let $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5\}$ denote the ordered set of pipeline stages, defined as follows:

$$c_1: \quad \texttt{preprocess} \quad f_{\text{pre}}: \mathcal{X}_{\text{raw}} \to \mathcal{X}, \quad \mathbf{x} \mapsto \tilde{\mathbf{x}} \tag{1}$$

$$c_2: \quad \texttt{tune} \quad f_{\text{tune}}: \Lambda \to \Lambda^*, \quad \lambda^* = \arg\min_{\lambda \in \Lambda} \mathcal{L}(f_{\text{pred}} \circ f_{\text{train}}(\lambda)) \tag{2}$$

$$c_3: \quad \texttt{train} \quad f_{\text{train}}: \mathcal{X} \times \Lambda^* \to \mathcal{H}, \quad (\tilde{\mathbf{x}}, \lambda^*) \mapsto h^* \tag{3}$$

$$c_4: \quad \texttt{predict} \quad f_{\text{pred}}: \mathcal{X} \to \mathcal{Y}, \quad \mathbf{x} \mapsto \hat{y} = h^*(\mathbf{x}) \tag{4}$$

$$c_5: \quad \texttt{score} \quad f_{\text{score}}: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}, \quad (\hat{y}, y) \mapsto \ell \tag{5}$$

where $\mathcal{X}_{\text{raw}}$ is the raw input space, $\mathcal{X}$ the preprocessed feature space, Λ the hyperparameter search space, Λ^* the optimal hyperparameter configuration, $\mathcal{H}$ the hypothesis class, and $\mathcal{Y}$ the label space. Stage c_2 is implemented as a Ray Tune orchestration loop that repeatedly calls f_{train} and f_{pred} until the trial budget $B \in \mathbb{N}$ is exhausted, returning $\lambda^* = \arg\min_\lambda \mathcal{L}$. Stages c_3–c_5 are then executed once with λ^*. Each stage $c \in \mathcal{C}$ incurs a measurable energy cost:

$$E^{(c)}(r) \in \mathbb{R}_+,$$

where r denotes a run and energy is reported in kWh (carbon footprint in gCO_2eq).

Setup and Notation. Let S denote the source domain, $\{T_1, \ldots, T_N\}$ a set of N target domains, and $c \in \mathcal{C}$ a pipeline stage. The total cost of a strategy π over stage c and N target domains is denoted $C^{(c)}_{\pi,\text{total}}(N)$. Three strategies are compared:

- **Independent Classification (IC):** for each target domain T_i, all stages $c \in \mathcal{C}$ are executed independently, including labeling and full retraining from scratch.
- **UDA – Sequential Adaptation (UDA-S):** stages c_1–c_3 are executed once on S; stages c_1 and c_3 (adaptation) are then repeated independently for each T_i, $i = 1, \ldots, N$.
- **UDA – Joint Multi-Target Adaptation (UDA-J):** stages c_1–c_3 are executed once on S; a single adaptation run at stage c_3 is performed jointly over $\{T_1, \ldots, T_N\}$ simultaneously.

Independent Classification Cost Model. In the independent setting, stage c_1 incurs both a labeling cost and a computational cost, while all other stages incur only computational cost. The total per-domain energy cost at stage c

is:

$$E_{\text{ind}}^{(c)} = \begin{cases} E_{\text{label}}^{(c_1)} + E_{\text{comp}}^{(c_1)} & \text{if } c = c_1, \\ E_{\text{comp}}^{(c)} & \text{otherwise,} \end{cases}$$

where the average computational cost per target domain at stage c is estimated as:

$$E_{\text{comp}}^{(c)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{r \in \mathcal{R}_{T_i,c}} E^{(c)}(r),$$

with $\mathcal{R}_{T_i,c}$ the set of runs executed for target domain T_i at stage c, and $E^{(c)}(r)$ the measured energy for run r.

The total cost over N target domains thus scales linearly:

$$E_{\text{IC,total}}^{(c)}(N) = E_{\text{source}}^{(c)} + N \cdot E_{\text{ind}}^{(c)},$$

UDA Cost Model: Sequential Adaptation (UDA-S). The source model is trained once and adapted independently to each T_i. No labeling cost is incurred at any stage. The total cost is:

$$E_{\text{UDA-S,total}}^{(c)}(N) = E_{\text{source}}^{(c)} + N \cdot E_{\text{adapt}}^{(c)},$$

where:

$$E_{\text{source}}^{(c)} = \sum_{r \in \mathcal{R}_{s,c}} E^{(c)}(r)$$

$$E_{\text{adapt}}^{(c)} = \frac{1}{|\mathcal{R}_{\text{UDA},c}|} \sum_{r \in \mathcal{R}_{\text{UDA},c}} E^{(c)}(r).$$

UDA Cost Model: Joint Adaptation (UDA-J). The source model is adapted once to all N target domains simultaneously. The total cost reduces to:

$$E_{\text{UDA-J,total}}^{(c)}(N) = E_{\text{source}}^{(c)} + E_{\text{adapt,joint}}^{(c)}(N),$$

where $E_{\text{adapt,joint}}^{(c)}(N)$ is the energy cost of a single joint adaptation run over all N targets. Since all domains are processed in a single pass:

$$E_{\text{adapt,joint}}^{(c)}(N) \le N \cdot E_{\text{adapt}}^{(c)},$$

with equality only when the joint run degenerates to N sequential passes.

Break-Even Analysis. The break-even number of targets N^* is the minimum N beyond which a UDA strategy becomes more energy-efficient than IC.

UDA-S break-even:

$$N_{\text{S}}^* = \frac{E_{\text{source}}^{(c)}}{E_{\text{ind}}^{(c)} - E_{\text{adapt}}^{(c)}}, \qquad \text{provided } E_{\text{adapt}}^{(c)} < E_{\text{ind}}^{(c)}.$$

UDA-J break-even:

$$N_{\text{J}}^* = \frac{E_{\text{source}}^{(c)} + E_{\text{adapt,joint}}^{(c)}}{E_{\text{ind}}^{(c)}}.$$

Since $E_{\text{adapt,joint}}^{(c)}$ is fixed regardless of N, it follows that $N_{\text{J}}^* \le N_{\text{S}}^*$, confirming that joint adaptation reaches the efficiency threshold earlier. The three strategies are summarized in Table 1.

Table 1. Summary of total energy cost models and break-even thresholds. (*) For IC, the $(N + 1)$ factor accounts for the source domain.

Strategy	Total Cost $E^{(c)}$	Scales with N	Break-Even N^*
IC*	$(N+1) \cdot (E_{\text{label}}^{(c_1)} + E_{\text{comp}}^{(c)})$	Linear	—
UDA-S	$E_{\text{source}}^{(c)} + N \cdot E_{\text{adapt}}^{(c)}$	Linear	$\frac{E_{\text{source}}^{(c)}}{E_{\text{ind}}^{(c)} - E_{\text{adapt}}^{(c)}}$
UDA-J	$E_{\text{source}}^{(c)} + E_{\text{adapt,joint}}^{(c)}$	Sublinear	$\frac{E_{\text{source}}^{(c)} + E_{\text{adapt,joint}}^{(c)}}{E_{\text{ind}}^{(c)}}$

4. Experiments

4.1. Datasets

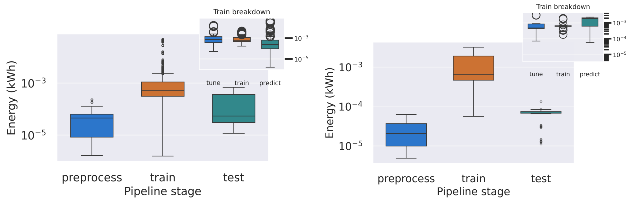
To the best of our knowledge, no open-source dataset exists for UDA-based time series classification in the wireless or telecommunications domain. We therefore follow Ismail Fawaz et al. (2025) and select benchmarks from adjacent domains exhibiting the distribution shifts and label scarcity characteristic of real-world wireless deployments, spanning three application areas. **Human activity recognition:** HAR (Anguita et al., 2013), HHAR (Stisen et al., 2015), and WISDM (Kwapisz et al., 2011) provide multivariate wearable sensor signals (accelerometers, gyroscopes) with the task of inferring activity type. **Sports and daily-life motion:** five body-worn sensors record 19 activities from 8 subjects (Altun et al., 2010), with standard windowing, denoising, and normalization applied. **Mechanical fault diagnosis:** the CWRU bearing dataset (bea) contains vibration signals from motor bearings under normal and faulty conditions, preprocessed via detrending, band-pass filtering, and segmentation.

4.2. Models and experimental setup

The experimental setup used here is closely related to the UDA benchmark performed in (Ismail Fawaz et al., 2025), where the following UDA approaches representative of current practice are compared.

Adversarial representation alignment: methods that train a feature extractor to produce domain-invariant representations by adversarial confusing a domain discriminator.

Discrepancy minimization: methods that minimize explicit measures of distribution mismatch (for example, kernel-



(a) UDA *(b)* Independent classification

Figure 2. Per-stage energy consumption (kWh) on a logarithmic scale for the UDA pipeline (*left*) and the independent classification pipeline (*right*). Each box reports the distribution over all experimental runs. The inset details the breakdown of the training stage into its three sub-components: hyperparameter search (`tune`), final model fitting (`train`), and inference (`predict`).

based distances such as maximum mean discrepancy) or that employ specialized convolutional/encoder backbones designed for time series.

Strong representation learners: encoder architectures (e.g., Inception-like encoders) used in combination with adaptation objectives; these often attain high accuracy but can be computationally more demanding.

Task-independent classification baseline: Since the Inception architecture consistently emerged as the strongest backbone across UDA experiments (Ismail Fawaz et al., 2025), InceptionTime was selected as the task-independent baseline.

For each run, energy consumption is measured using CodeCarbon (Courty et al., 2024a), and all cost estimates are reported as empirical means over MLflow-tracked runs. All experiments are implemented in PyTorch and executed using the Ray framework for distributed hyperparameter tuning and experiment orchestration. To ensure a fair comparison across all methods, we adopt a fixed time budget for both the tuning stage (c_2) and the training stage (c_3), held constant across all models and datasets. This budget-based protocol prevents any single method from benefiting from disproportionate tuning or training resources, and ensures that the reported energy costs are directly comparable across strategies.

4.3. Results

Energy consumption per stage. Figure 2 breaks down the estimated energy consumption by pipeline stage for (a) the UDA and (b) the independent classification settings. The training stage dominates by two to three orders of magnitude over preprocessing and inference. This is explained by the pipeline structure: the tuning phase (`tune`) repeatedly calls `train` and `predict` until the hyperparameter budget is exhausted, after which a final `train` run is performed with the best configuration found, followed by a single `score` pass. Consequently, `tune` concentrates the bulk of the total energy and drives most of the run-to-run variance, while the final `train` and `score` steps contribute a comparatively negligible share. The UDA pipeline exhibits a higher upper tail than the classification setting, reflecting the sensitivity of distribution-matching objectives. Whether UDA or independent classification is the greener choice overall cannot be determined from this view alone.

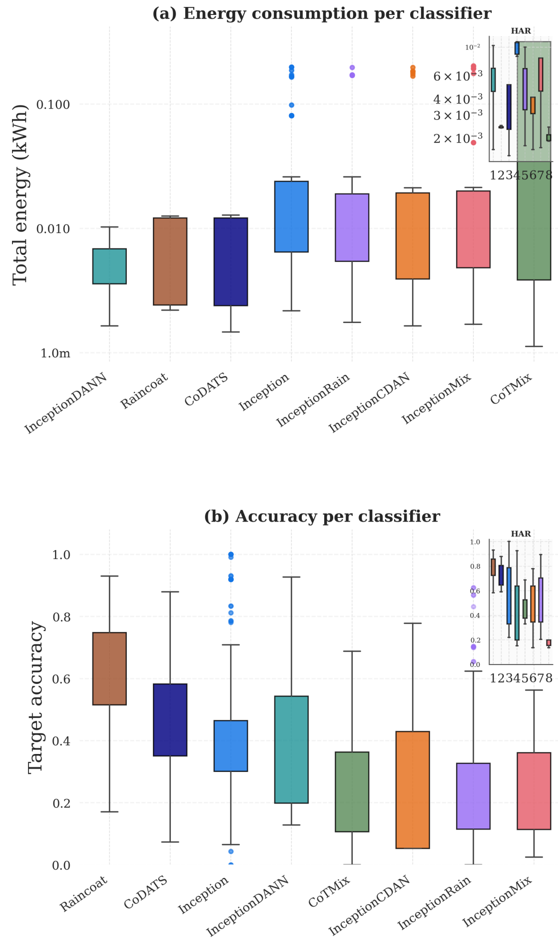


Figure 3. Comparison of energy consumption (KWh, top) and performance (bottom) across algorithms.

Energy consumption per algorithm. Figure 3 jointly characterizes the energy-accuracy profile of UDA classifiers. Figure (a) shows that InceptionDANN, Raincoat, and CoDATS remain the cheapest group (median below 10^{-2} kWh), while Inception-based UDA variants consume roughly 15–20 m kWh at median. CoTMix stands apart with an inter-quartile range spanning nearly two orders of magnitude, reflecting highly variable convergence across transfer pairs. The upper right plots show a zoom for the HAR dataset, which confirms that these relative rankings are stable, with all models.

Figure (b) shows a relative downward shift in accuracy com-

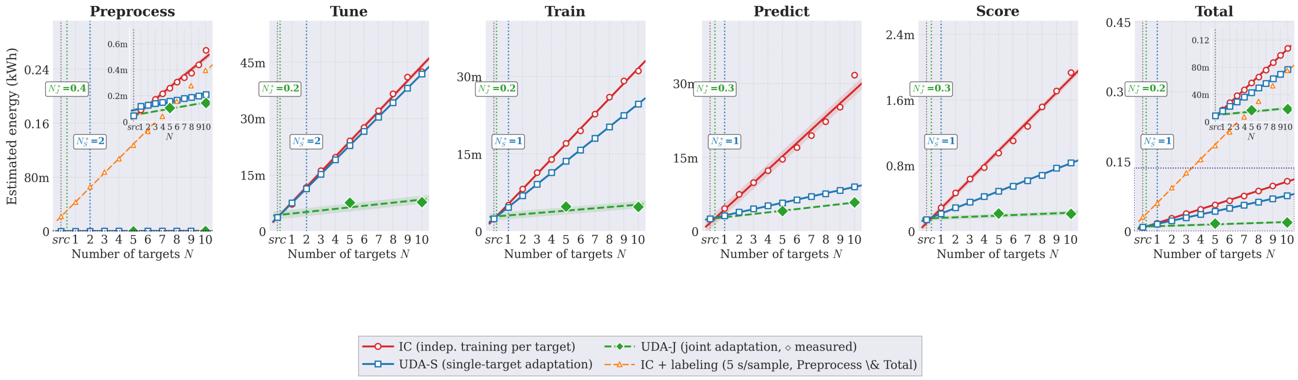


Figure 4. Estimated pipeline energy as a function of the number of target domains N on the HAR dataset, broken down by pipeline stage and aggregated (Total). **IC** trains one model per target independently; **UDA-S** adapts a shared source model sequentially; **UDA-J** performs a single joint adaptation (◇: measured at $N = 5, 10$).

pared to Fawaz et al. (2023), benchmark we adopt in this study. We attribute this gap to our reduced computational budget: whereas Fawaz et al. (2023) uses up to thousands of tuning runs per model, our protocol is constrained to 5 experiment sets within 15–400 tuning and training budget range. This reflects a realistic deployment constraint and is itself a contribution of the study, but inevitably penalises tuning-sensitive models such as CoTMix and InceptionMix, whose wide interquartile ranges confirm high variance across runs. Overall, the energy-accuracy trade-off under our budget is best represented by Raincoat (low cost, robust accuracy) and InceptionRain (moderate cost, best accuracy among Inception-based UDA methods). The rest of our analysis focuses on these two UDA algorithms and InceptionTime for the IC setting.

Identifying the break-even phenomenon. Figure 4 addresses the central question of this work: *when does UDA become the greener choice?* Each panel plots the estimated $E^{(c)}(N)$ as a function of the number of target domains N for the three strategies: IC (red), UDA-S (blue), and UDA-J (green). The orange lines denoted as IC+labeling on the preprocessing and total panels correspond to $E^{(c_1)}_{\text{label}}$, which accounts for the labeling cost. Across all stages, the break-even values are remarkably low. UDA-J reaches its threshold N^*_{J} at $N = 1.0$-1.4 targets depending on the stage, meaning that adapting to a joint target domain is already sufficient to amortise the source training cost $E^{(c)}_{\text{source}}$. UDA-S breaks even slightly later at $N^*_{\text{S}} = 1$–2. Beyond these thresholds, both UDA strategies scale far more favourably than IC: UDA-S grows linearly but with a slope $E^{(c)}_{\text{adapt}} \ll E^{(c)}_{\text{ind}}$, while UDA-J remains nearly flat as $E^{(c)}_{\text{adapt,joint}}$ is fixed regardless of N. The Tune stage confirms that hyperparameter search is the dominant cost driver across all strategies. Both IC and UDA-S employ the same Hydra-based tuning protocol with identical budget settings, so their per-target tuning costs are comparable and both scale linearly with N.

The preprocessing and total panels show that once annotation cost $E^{(c_1)}_{\text{label}}$ is included, IC+labeling (orange) diverges steeply from the computational-only IC curve (red), making UDA strictly dominant from $N \geq 1$ in practical scenarios where target labels are unavailable or expensive to obtain. Under our experimental protocol, UDA becomes the greener choice from as few as $N^* = 1$-2 target domains across every pipeline stage, and the advantage compounds rapidly with N. When labeling cost is factored in, UDA is strictly preferable from the very first target domain.

5. Conclusion

Task-specific factors in telecom settings, including labeling effort, dataset size, and cross-site heterogeneity, shape scenarios in which UDA offers a clear technical advantage. However, the energy efficiency of UDA is not well studied, and it is important to evaluate it to ensure green communication networks. In this work, we present an energy–accuracy analysis of Unsupervised Domain Adaptation for time series classification, asking when UDA becomes the greener choice? Our break-even analysis shows that UDA reaches energy parity from as few as one to two target domains, and is strictly preferable from the first target once labeling cost is accounted for. Joint multi-target adaptation should be preferred whenever feasible. Future work should extend evaluations to include more diverse datasets, develop unsupervised validation metrics to reduce tuning cost, and refine cost models under realistic labeling assumptions.

References

CWR bearing dataset. Case Western Reserve University. URL https://engineering.case.edu/bearingdatacenter/welcome.

Altun, K., Barshan, B., and Tunçel, O. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.

Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. A public domain dataset for human activity recognition using smartphones. In *ESANN*, volume 3, pp. 3, 2013.

Anthony, L. F. W., Kanding, B., and Selvan, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*, 2020.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

Bothe, S., Saffar, I., Boisbunon, A., Farooq, H., Forgeat, J., and Chowdhury, M. M. U. Through the telecom lens: Are all training samples important? *arXiv preprint arXiv:2511.21668*, 2025.

Budennyy, S. A., Lazarev, V. D., Zakharenko, N. N., Korovin, A. N., Plosskaya, O. A., Dimitrov, D. V., Akhripkin, V. S., Pavlov, I. V., Oseledets, I. V., Barsola, I. S., et al. Eco2ai: carbon emissions tracking of machine learning models as the first step towards sustainable ai. In *Doklady mathematics*, volume 106, pp. S118–S128. Springer, 2022.

Buzzi, S., Chih-Lin, I., Klein, T. E., Poor, H. V., Yang, C., and Zappone, A. A survey of energy-efficient techniques for 5g networks and challenges ahead. *IEEE Journal on selected areas in communications*, 34(4):697–709, 2016.

Courty, B., Schmidt, V., Luccioni, S., Goyal-Kamal, Coutarel, M., Feld, B., Lecourt, J., Connell, L., Saboni, A., Inimaz, supatomic, Léval, M., Blanche, L., Cruveiller, A., ouminasara, Zhao, F., Joshi, A., Bogroff, A., de Lavoreille, H., Laskaris, N., Abati, E., Blank, D., Wang, Z., Catovic, A., Alencon, M., Stęchły, M., Bauer, C., de Araújo, L. O. N., JPW, and Books, M. mlco2/codecarbon: v2.4.1, May 2024a. URL https://doi.org/10.5281/zenodo.11171501.

Courty, B., Schmidt, V., Luccioni, S., Goyal-Kamal, MarionCoutarel, Feld, B., Lecourt, J., LiamConnell, Saboni, A., Inimaz, supatomic, Léval, M., Blanche, L., Cruveiller, A., ouminasara, Zhao, F., Joshi, A., Bogroff, A., de Lavoreille, H., Laskaris, N., Abati, E., Blank, D., Wang, Z., Catovic, A., Alencon, M., Stęchły, M., Bauer, C., de Araújo, L. O. N., JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024b. URL https://doi.org/10.5281/zenodo.11171501.

Desislavov, R., Martínez-Plumed, F., and Hernández-Orallo, J. Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38:100857, 2023.

Ericsson. When AI has no time to think. *Ericsson Technology Review*, 2023a. URL https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/when-ai-has-no-time-to-think.

Ericsson. Ericsson sustainability and corporate responsibility report. Technical report, Ericsson, 2023b. URL https://www.ericsson.com/4931d8/assets/local/about-ericsson/sustainability-and-corporate-responsibility/documents/2023/modern-slavery-and-human-trafficking-statement.pdf.

Ericsson. Ericsson mobility report. Technical report, Ericsson, November 2025. URL https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast.

Fawaz, H. I., Del Grosso, G., Kerdoncuff, T., Boisbunon, A., and Saffar, I. Deep unsupervised domain adaptation for time series classification: a benchmark. *arXiv preprint arXiv:2312.09857*, 2023.

Ismail Fawaz, H., Del Grosso, G., Kerdoncuff, T., Boisbunon, A., and Saffar, I. Deep unsupervised domain adaptation for time series classification: a benchmark: Hi fawaz et al. *Data Mining and Knowledge Discovery*, 39 (4):39, 2025.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Lai, N., Dewi, D. A., Maidin, S. S., Xiao, W., Zhao, S., and Hu, Q. A comprehensive review of lightweight deep learning models for edge computing with future directions. *Discover Computing*, 29(1):110, 2026.

Lin, J., Zhu, L., Chen, W.-M., Wang, W.-C., Gan, C., and Han, S. On-device training under 256kb memory. *Advances in Neural Information Processing Systems*, 35: 22941–22954, 2022.

Maatouk, A., Piovesan, N., Ayed, F., De Domenico, A., and Debbah, M. Large language models for telecom: Forthcoming impact on the industry. *IEEE Communications Magazine*, 63(1):62–68, 2024.

Malmodin, J., Lövehagen, N., Bergmark, P., and Lundén, D. Ict sector electricity consumption and greenhouse gas emissions–2020 outcome. *Telecommunications Policy*, 48(3):102701, 2024.

Mao, B., Tang, F., Kawamoto, Y., and Kato, N. Ai models for green communications towards 6g. *IEEE Communications Surveys & Tutorials*, 24(1):210–247, 2021.

Next G Alliance Green G Working Group. Sustainable ai in telecom: Promises and challenges in 6g. White paper, ATIS Next G Alliance, February 2025. URL https://nextgalliance.org/white_papers/sustainable-ai-in-telecompromises-and-challenges-in-6g/.

O'shea, T. and Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Raghuram, J., Zeng, Y., Martí, D. G., Ortiz, R. R., Jha, S., Widmer, J., and Banerjee, S. Few-shot domain adaptation for end-to-end communication. *arXiv preprint arXiv:2108.00874*, 2021.

Raza, H., Prasad, G., and Li, Y. Adaptive learning with covariate shift-detection for non-stationary environments. In *2014 14th UK Workshop on Computational Intelligence (UKCI)*, pp. 1–8. IEEE, 2014.

Saffar, I., Morel, M. L. A., Singh, K. D., and Viho, C. Semi-supervised deep learning-based methods for indoor outdoor detection. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2019.

Soares, J. A., Mayer, K. S., and Arantes, D. S. Semi-supervised ml-based joint channel estimation and decoding for m-mimo with gaussian inference learning. *IEEE Wireless Communications Letters*, 12(12):2123–2127, 2023.

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pp. 127–140, 2015.

Strubell, E., Ganesh, A., and McCallum, A. 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, 2019.

Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., and Sidiropoulos, N. D. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018.

Talak, R., Karaman, S., and Modiano, E. Optimizing information freshness in wireless networks under general interference constraints. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 61–70, 2018.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of machine learning and systems*, 4:795–813, 2022.