Renewable Energy, Environment, and Power Quality Journal (REE&PQJ)
ISSN:3020-531X
Volume 26, July 2026

Leakage-Safe Evaluation for Day-Ahead Load Forecasting Under an Operational Data Gap Constraint based on a Minimal Information Set

Luis Carreño-Barrera∗, Jairo Blanco-Solano∗, Cesar Duarte∗
∗School of Electrical, Electronic and Telecommunications Engineering (E3T)
Universidad Industrial de Santander (UIS)
Bucaramanga, Colombia

Emails: luis.carreno@e3t.uis.edu.co, jablanso@uis.edu.co, cedagua@uis.edu.co

Abstract. **End-to-end pipelines with data leakage can mislead model selection in day-ahead load forecasting. When training and evaluation inadvertently violate the information set available at issuance time, performance estimates cease to be decision-relevant. We quantify this effect by contrasting a leakage-safe (deployment-like) protocol against common leaky evaluation pitfalls for three base learners: Ridge regression, Light Gradient Boosting Machine (LightGBM), and a Long Short-Term Memory (LSTM) network. We adopt a forecast-free information set that enforces an operational delay (gap) of g = 18 h at issuance time t0; i.e., models only use observations available up to t0 – g together with deterministic calendar descriptors (hour-of-day, day-of-week, holidays) known at issuance time. Importantly, we intentionally restrict inputs to this minimal, deployment-robust feature set (load history + calendar only, no exogenous variables or forecasts), yet still obtain competitive accuracy. Load forecasts are evaluated through a deployment-like day-ahead simulator with a fixed daily issuance time, and learning-based models are trained in residual form relative to a Naive-week (weekly persistence) baseline model. Across three Colombian served-load datasets—the national interconnected bulk power system and two distribution system operator (DSO) electricity markets—split contamination can change conclusions and inflate accuracy. Under the proposed leakage-safe protocol on the medium-sized DSO market, Ridge yields the lowest mean absolute percentage error (MAPE), 2.13%, among the three learners, while leaky split contamination can make LightGBM appear unrealistically strong (MAPE = 0.64%). In contrast, Ridge is comparatively leakage-robust, showing only marginal changes between leakage-safe and contaminated evaluations, unlike higher-capacity learners. Despite using only load-and-calendar inputs, Ridge also matches or surpasses the DSO market forecast across the studied areas. We additionally compute mean absolute error (MAE) and root mean squared error (RMSE) to complement MAPE.**

Key words. **Day-ahead load forecasting; leakage-safe evaluation; forecast-free information set; operational gap; data leakage.**

1. INTRODUCTION

Short-term day-ahead (H=24) load forecasts support operational planning, reserve sizing, and market procedures [1]–[3]. Planning and operational reports motivate reproducible and auditable forecasting workflows, where methodology and evaluation must be consistent with real deployment constraints [4]–[7].

Modern power-system forecasting studies face three simultaneous requirements. First, deployment realism: forecasts are issued at a defined time, and only past observations are available, often with an operational gap before the first predicted hour (horizon). Second, robust evaluation: the test set must lie strictly in the future relative to the training and validation sets; otherwise, temporal data leakage is introduced [8]-[10]. Third, strong baselines: pronounced weekly seasonality makes last-week same-hour a hard benchmark in practice [11]. From a feature engineering perspective, competitive short-term load forecasting often relies on a small set of robust input groups—lagged load history together with deterministic calendar descriptors (hour-of-day, day-of-week, holidays) and seasonal terms—which remain effective across model families, from semiparametric formulations to tree-based models [1]–[3]. Recent deployment-oriented studies with gradient-boosted trees likewise emphasize calendar and autoregressive features under rolling, time-respecting evaluation [12], [13]. We therefore treat the specification of the information set and the choice of strong seasonal benchmarks as first-order methodological decisions. However, many machine-learning studies still omit one or more of these requirements: preprocessing is sometimes performed over the whole information set. Time splits for the information set are defined after devising input descriptors, gap and forecast horizon (windowing), and evaluation may inadvertently reuse information unavailable at issuance time (data leakage) [14]. These failures are often unintentional: common notebook-style workflows make it easy to apply global preprocessing or to generate sliding windows before enforcing strict time splits, and subtle boundary contamination can go unnoticed without explicit leakage-aware checks [9], [15]. These risks are amplified in multi-horizon forecasting, where overlapping labels make leakage easier to introduce, motivating leakage-aware diagnostics and leakage-safe preprocessing variants [16]–[18].

Despite the operational criticality of day-ahead load forecasting, a key unmet need remains: leakage-safe,

deployment-like evaluation protocols that yield decision-relevant performance estimates [8], [19]. Many studies (i) implicitly assume information that is unavailable at issuance (e.g., ex-post access to future exogenous variables), (ii) under-specify leakage control in multi-horizon settings where overlapping labels make contamination easy to introduce [9], [16], or (iii) benchmark against weak seasonal baselines despite strong weekly persistence in practice [11]. As a result, reported performances can be inflated and may not translate to deployment performance under true issuance constraints [14], [16]. This paper addresses day-ahead load forecasting under a strictly forecast-free information set with an operational gap of g=18 hours. At issuance t0, features are built only from observations available up to t0−g, together with deterministic calendar descriptors for the target timestamps (known at issuance without forecasting). We focus on an interpretable comparison of three widely used base learners—LightGBM (LGBM), Ridge Regression (Ridge), and a long short-term memory network (LSTM)—trained in residual form relative to the Naive-week (weekly persistence) baseline model. We report results under two evaluation regimes: (i) a leakage-safe, deployment-like protocol, and (ii) a leaky protocol that mimics common antipatterns (e.g., training contamination by the test set), to quantify how model ranking can change. Our contribution is methodological about which testing evidence is valid for model ranking and selection.

A. *Contributions*

1) *Leakage-safe, deployment-like testing under an operational gap:* we formalize a day-ahead (H=24) served-load forecasting setup with a forecast-free information set and an explicit issuance-time gap (g=18 h), and implement an issuance simulator (deployment-like) that enforces these constraints end-to-end.

2) A strategy to quantify how leakage distorts model selection: we construct an end-to-end split-contamination pipeline representative of common time-series pitfalls and show that it can yield inflated accuracy and distort the ranking of competing learners, making performance estimates non-decision-relevant for deployment.

3) Minimal information set and descriptor design with competitive residual forecasting: we adopt a minimal, deployment-robust input design—served-load history together with deterministic calendar descriptors known at issuance time—and train residual models relative to a strong Naive-week baseline, achieving competitive accuracy that matches or outperforms forecasts issued by DSOs.

The rest of the paper is organized as follows. Section **2** defines the forecast-free formulation with an operational gap. Section **3** describes the methodology. Sections **4** and **5** report the experimental setup and results. Section **6** presents conclusions and future work.

2. PROBLEM STATEMENT

A. *Forecast-free minimal information set with an operational gap*

We study day-ahead forecasting with a horizon length of $H = 24$ (one full delivery day). Forecasts are issued at time t0 for the next H hours. Due to operational constraints (e.g., data latency and daily update deadlines), the last usable measurements at issuance are those observed up to $t_0 - g$, where g is the operational information gap. In the Colombian operational setting, day-ahead schedules are updated until approximately 07:00, leaving an ≈ 18 to $19h$ gap to the start of the forecasted period (H) [4]. We use $g = 18$ h as a conservative deployment proxy. We fix the short-term look-back length to $L = 24\,h$, whose served load, y_t (MW) for $t_0 - g - L + 1 \le t \le t_0 - g$ hours, is used as descriptors for the models. Calendar information at hour $t, denoted\ by\ c_t$, is deterministic and thus available at issuance time for both the historical window and the forecast horizons; therefore, we include $c_{t0} - g - L + 1 : t_0 - g$ and $c_{t0} : t_0 + H - 1$ in the information set $\mathcal{I}_{t_0}$.

We enforce a forecast-free (minimal) information set and no exogenous measurements or forecasts are used as model inputs. At issuance time, features are constructed only from historical load observations and deterministic calendar descriptors for the target timestamps (e.g., hour-of-day, day-of-week, holidays), which are known at issuance without forecasting. Formally, after the described windowing process, the admissible information set is

$$\mathcal{I}_{t_0} = \begin{Bmatrix} y_{t_0-g-L+1:t_0-g},\ y_{t_0+1-168:t_0+H-168}, \\ c_{t_0-g-L+1:t_0-g},\ c_{t_0:t_0+H-1} \end{Bmatrix}. \qquad (1)$$

B. Direct multi-horizon forecasting

We consider direct multi-horizon forecasting at once, thus: for each horizon h $\in$ {1, . . ., H − 1}, a model delivers the predicted demand:

$$\hat{y}_{t_0+h} = f\big(\phi(\mathcal{I}_{t_0}), h\big), \qquad h = 1,2,3,\dots,H. \qquad (2)$$

where $\phi(\cdot)$ is a causal feature mapping constructed from $\mathcal{I}_{t_0}$. Depending on the model, $f(\cdot)$ can be implemented as (i) one model per horizon or (ii) a multi-output regressor; in all cases, performance is evaluated per horizon and also aggregated over the full horizon. All three models were trained for direct multi-horizon forecasting under the forecast-free, gap-constrained proposed protocol.

C. Naive-week (weekly persistence) baseline model

As a hard, deployment-relevant comparator, we define the Naive-week (weekly persistence) forecast, which repeats the observed load from the same hour one week earlier, for each horizon [11]:

$$\hat{y}_{t_0+h}^{(\text{naive})} = y_{t_0+h-168} \qquad (3)$$

This baseline model is feasible under the gap g because $t_0 + h - 168 \leq t_0 - g$ for $h \leq 24$ and $g = 18$. When used as an input/anchor for learning-based models (e.g., residual learning), Naive-week is part of the admissible information set since it depends only on past observations available under the gap.

D. Residual learning relative to Naive-week

To explicitly test whether learning-based methods add value beyond Naive-week persistence, we adopt a residual formulation. The weekly residual at horizon h is

$$r_{t_0+h} = y_{t_0+h} - \hat{y}^{(\text{naive})}_{t_0+h} \quad (4)$$

Then, models are trained to predict residual demand relative to the weekly baseline and the forecasted demand is reconstructed as

$$\hat{y}^{(m)}_{t_0+h} = \hat{y}^{(\text{naive})}_{t_0+h} + \hat{r}^{(m)}_{t_0+h} \quad (5)$$

where $\hat{r}^{(m)}_{t_0+h}$ is the residual prediction of model $m \in \{\text{LGBM,Ridge,LSTM}\}$. Thus, each model anchors every horizon to the weekly seasonal value $y_{t_0+h-168}$ (previous week same-hour) and learns only the residual correction.

E. Evaluation objective

We prioritize MAPE as the main operational metric and also report MAE and RMSE, aggregated across the full horizon and evaluated through a day-ahead issuance simulator.

3. METHODOLOGY

We compare three models under a fixed, deployment-oriented information set and quantify how common leakage antipatterns can inflate reported accuracy.

A. Forecast-free supervised formulation with an operational gap

At each issuance time t_0, forecasts are produced under a forecast-free (minimal) information set with an explicit operational delay $g = 18$ h. Models are allowed to use only information observed up to $t_0 - g$; any feature that would require future observations is excluded. Supervised samples are built from a rolling window of past observations and deterministic time descriptors:

$$x_{t_0} = \phi\begin{pmatrix} y_{t_0-g-L+1:t_0-g},\ y_{t_0+1-168:t_0+H-168}, \\ c_{t_0-g-L+1:t_0-g},\ c_{t_0:t_0+H-1} \end{pmatrix} \quad (6)$$

$$\hat{y}_{t_0+h} = f(x_{t_0}, h), \qquad h = 1,2,\dots,H. \quad (7)$$

for horizons $h \in \{1,\dots,H\}$. No exogenous measurements or forecasts (including weather forecasts) are used as model inputs.

Feature engineering transparency. In the reported experiments, input features are derived exclusively from past load observations and deterministic time descriptors. Concretely, $\phi(\cdot)$ includes: (i) the $L = 24$ h lagged load history together with causal persistence terms (24 h and 168 h) and backward-looking rolling summaries computed using only past observations (implemented via one-step shifts), (ii) deterministic calendar descriptors computed for the historical window timestamps (hour-of-day, day-of-week, day-of-year, and holiday related flags), and (iii) train-only fitted encodings: an empirical CDF transform of the target fitted on the training block and Fourier sin / cos terms using periods selected from the training block via FFT and then applied unchanged to validation/test. No temperature, weather, or other exogenous variables are used. All learning models are trained in residual mode relative to the Naive-week baseline model (Section **2**).

B. Evaluation protocols: leakage-safe vs. leaky

Leakage-safe protocol (deployment-like). We define a strict chronological split into **train**, **validation**, and **test** by timestamp cutoffs [9]. Preprocessing (e.g., target scaling) is fitted on training set only and then applied to validation and testing sets. Supervised windows are constructed within each split block, so that no training sample contains labels from a future test period. Final performance is computed on the held-out test period via a deployment-like day-ahead simulator (Section **4**). This split-before-windowing design induces a natural purge/embargo at each split boundary: samples near a boundary are dropped if their look-back window or any of their H future labels would cross into the next block, preventing overlap-driven leakage [10], [16].

Model selection and tuning. All model configurations are selected using the validation set only; the testing set is used once for final reporting under the issuance-time simulator.

Leaky protocols (comparison purposes). To quantify how performances can be inflated, we devise leaky regimes that mimic common antipatterns highlighted in methodological guidelines for leakage-robust load forecasting: (i) training contamination where the training set accidentally includes part of the intended test window (e.g., splitting after windowing), and (ii) global preprocessing where scalers or summary statistics are fit on the full series before splitting [14], [16]. Leaky pipelines are diagnostics only; model selection is restricted to the proposed leakage-safe protocol. We distinguish two leaky variants:

• *Leaky-S (split contamination; inflation-prone):*

The test window spans Feb–Aug 2024. TRAIN is contaminated by including the full test window (Feb–Aug 2024), and accuracy is (incorrectly) reported on that same window. Target scaling uses *sklearn.preprocessing.MinMaxScaler* fitted on the contaminated pool (TRAIN plus Feb–Aug 2024) and applied to the full series.

• *Leaky-F (full leak; invalid and potentially degrading):*

In addition to Leaky-S contamination and target scaling, we apply operations that use information from future timestamps by fitting preprocessing on the full series before the train/test split. Concretely: (i) missing values are filled using both past and future observations within a 24 h window, (ii) clipping thresholds (at the 0.5% and 99.5% quantiles) are computed on the full series, (iii) the empirical CDF transform and FFT-based seasonal periods are fitted on the full series, and (iv) we add near-future covariates derived from the target, including a 24 h centered rolling mean and a 1 h negative shift. The issuance-time simulator ignores the operational gap by setting the history cutoff to the forecast start time. TRAIN is contaminated by including the full test window (Feb–

Aug 2024), and accuracy is (incorrectly) reported on that same window.

4. EXPERIMENTAL SETUP

A. Forecasting protocol

We study hourly multi-horizon served-load forecasting in the day-ahead setting defined in Section II. We evaluate forecasts through a deployment-like day-ahead simulation: we issue one forecast every 24 h at a fixed anchor hour (00:00) and score the subsequent non-overlapping 24 h block under the same gap-constrained, forecast-free information set. In the Colombian operational setting, day-ahead schedules are updated until approximately 07:00, leaving an ≈18 to 19 h information gap to the start of the delivery day; we use g=18 h as a conservative deployment proxy.

B. Data access (SIMEM)

We use three Colombian served-load datasets: the national interconnected bulk power system (SIN) and two DSO electricity markets, MC–Antioquia and MC–Santander (served load aggregated at the DSO market level for the Antioquia and Santander areas, respectively). All datasets are publicly available through the SIMEM REST endpoint https://www.simem.co/backend-files/api/PublicData. Each dataset is identified by a 6-character datasetId and queried by date range using startDate and endDate (YYYY-MM-DD); metadata and identifiers are available in the SIMEM catalog (datasetId=e007fb).

Served load is retrieved from the hourly dataset MC–Antioquia (target: DemandaAtendida) using datasetId=9e9ee7; the same dataset is used for MC–Santander and SIN by filtering the categorical market/system field via columnDestinyName and values (e.g., values=MC--Antioquia, values=MC--Santander, values=SIN), or by downloading the requested date range and filtering locally. For example, a one-day download uses startDate=endDate: https://www.simem.co/backend-files/api/PublicData?datasetId=9e9ee7&startDate=2026-02-05&endDate=2026-02-05&columnDestinyName=null&values=null.

C. Dataset and chronological split

We evaluate on three Colombian served-load series: two DSO electricity markets, *MC–Antioquia* and *MC–Santander*, and the national interconnected bulk power system aggregate, denoted *SIN*. The target is hourly served load (*DemandaAtendida*), reported in MW (converted from kW). *MC–Antioquia* represents 13.14% of *SIN*, and *MC–Santander* represents 3.73% of *SIN*. For all three series, the retained analysis period spans from 2022-01-08 00:00 to 2024-08-31 24:00; due to the gap/look-back feature-validity constraints, the first usable timestamp for supervised construction is 2022-01-08 01:00. We use a strict chronological split (70/10/20 train/validation/test), which yields cutoffs at 2023-11-15 21:00 (train end) and 2024-02-20 14:00 (validation end). The test window spans 2024-02-20 15:00 to 2024-08-31 24:00, covering the monthly evaluation windows Feb–Aug 2024 (plus a short trailing partial month).

Outlier clipping (train-only). To reduce the effect of extreme targets while preserving leakage safety, we clip *DemandaAtendida* using train-only quantiles (0.5%–99.5%) and apply the resulting bounds to the full series. The clipping ranges (in MW) are: *MC–Antioquia* [715.36, 1574.99], *MC–Santander* [252.48, 452.17], and *SIN* [7150.83, 11042.12].

D. Leakage-safe vs. end-to-end leaky pipelines

The leakage-safe protocol follows strict chronology: splits are defined by timestamps, all preprocessing is fit on the training block only, windows are generated within each block, and the test period is evaluated exclusively through the issuance-time simulator.

The end-to-end leaky pipelines intentionally violate these practices. We define a split-contamination variant (*Leaky-S*) that contaminates training with data from the evaluation period and fits preprocessing globally; performance is then incorrectly computed on that same contaminated evaluation period. We also include an aggressive variant (*Leaky-F*) that adds explicit future-peeking operations and ignores the operational gap, producing scores that are not decision-relevant under the forecast-free issuance-time information set. In both leaky variants, leakage is applied throughout the full evaluation period (i.e., no embargoed or untouched months are kept).

5. RESULTS

A. *Main comparison: leakage-safe vs. end-to-end leaky pipelines*

Table I summarizes test performance (*MAPE*) across three Colombian served-load datasets: two DSO electricity markets (*MC–Antioquia* and *MC–Santander*) and the national interconnected bulk power system aggregate (*SIN*). For each dataset, we report two baseline models (*Naive-week* and the *SIN* operator) and three residual learners (*Ridge*, *LGBM*, and *LSTM*) under two regimes: (i) the leakage-safe protocol (*Safe*) and (ii) an end-to-end leaky split-contamination pipeline (*Leaky-S*) that contaminates training with part of the test window and fits preprocessing globally. Across datasets, *Leaky-S* can change rankings: *LightGBM* appears unrealistically strong (e.g., *MAPE* = 0.64% on *MC–Antioquia*), whereas under the leakage-safe protocol *Ridge* yields the lowest *MAPE* among the three learners on each dataset (e.g., *MAPE* = 2.13% on *MC–Antioquia*) and is only slightly affected by split-contamination leakage.

Why does leakage differently affect models? (Qualitative interpretation)

Leakage effects can differ across model classes. *Ridge* is a regularized linear model and is typically less able to exploit complex, spurious patterns induced by split contamination or global preprocessing. In contrast, gradient-boosted trees such as *LightGBM* can fit high-capacity nonlinear rules that inadvertently latch onto leakage artifacts. *LSTMs* can be sensitive to distribution shifts and alignment choices, so leaky modifications may not consistently help.

Table II reports the same *Safe* vs. *Leaky-S* comparison using *MAE* and *RMSE* instead of *MAPE*. The conclusions are consistent across metrics: the regimes that reduce *MAPE* under *Leaky-S* also reduce *MAE/RMSE*, and the relative behavior of the learners across *Safe* vs. *Leaky-S* is preserved. This agreement indicates that the observed leakage effects are not specific to *MAPE*.

TABLE I: Test MAPE (%) under the leakage-safe protocol (Safe) vs. split-contamination (Leaky-S) across datasets. Δ denotes Safe minus Leaky-S (percentage points).

Dataset	Model	Safe	Leaky-S	Δ (pp)
MC–Antioquia DSO	Naive-week	4.45	–	–
	DSO market	2.99	–	–
	Ridge	2.13	2.03	+0.11
	LGBM	2.91	0.64	+2.27
	LSTM	5.10	6.58	−1.48
MC–Santander DSO	Naive-week	5.76	–	–
	DSO market	3.85	–	–
	Ridge	3.64	3.45	+0.19
	LGBM	4.16	0.81	+3.36
	LSTM	6.34	6.10	+0.24
SIN	Naive-week	3.13	–	–
	SIN operator	2.71	–	–
	Ridge	1.93	1.59	+0.33
	LGBM	2.30	0.45	+1.85
	LSTM	3.37	4.03	−0.67

B. Not all leakage increases accuracy

Leakage is often discussed as a source of optimistic backtests. However, under the full-leak setting (*Leaky-F*), test errors increase across datasets and learners. Table III reports *MAPE* under the leakage-safe protocol (*Safe*) versus *Leaky-F*; Δ (*Safe* minus *Leaky-F*) is negative for all learning models, indicating higher *MAPE* under *Leaky-F*. For example, on *MC–Antioquia*, *Ridge* increases from 2.13% (*Safe*) to 25.57% (*Leaky-F*), *LGBM* from 2.91% to 12.86%, and *LSTM* from 5.10% to 33.99%. The same direction holds on *MC–Santander* and *SIN*.

Table IV reports *Leaky-F* error levels using *MAE* and *RMSE* together with *MAPE*. The increases observed in *MAPE* under *Leaky-F* are also reflected in *MAE/RMSE* across datasets. For example, on *MC–Antioquia*, *Leaky-F* yields *MAE/RMSE* of 270.77/289.92 MW for *Ridge* and 136.12/158.30 MW for *LGBM*. When evaluation violates the information set available at issuance, performance estimates cease to be decision-relevant.

6. CONCLUSION

We studied day-ahead (H=24) served-load forecasting under a forecast-free information set with an operational gap (g=18 h), contrasting a leakage-safe (deployment-like) protocol with representative leaky evaluation pitfalls. When evaluation violates the information set available at issuance, performance estimates cease to be decision-relevant for deployment-facing model selection.

Key takeaways

– Methodology first: this paper is about which evaluation evidence is valid for model selection under issuance-time constraints.

– Protocol validity dominates conclusions: across datasets, split contamination can make LightGBM appear unrealistically strong (e.g., MAPE = 0.64%), whereas under the leakage-safe protocol Ridge yields the lowest MAPE among the three learners (e.g., MAPE = 2.13%).

– Ridge robustness to split-contamination leakage: Ridge is only marginally affected by Leaky-S, in contrast to higher-capacity learners whose rankings and reported accuracies can change under split contamination.

– Minimal information set with competitive accuracy: using only a minimal, deployment-robust descriptor design (load history + deterministic calendar features) and residual learning relative to Naive-week, Ridge matches or outperforms the current DSO market forecast across the studied datasets.

TABLE II: Test MAE/RMSE (MW, across the 24-hour horizon) under the leakage-safe protocol (Safe) vs. split contamination (Leaky-S) across datasets. Each cell reports MAE / RMSE.

Dataset	Model	Safe (MAE / RMSE)	Leaky-S (MAE / RMSE)
MC–Antioquia DSO	Naive-week	40.27 / 52.58	–
	DSO market	34.18 / 44.97	–
	Ridge	22.36 / 30.78	21.32 / 28.88
	LGBM	30.74 / 39.41	6.78 / 16.62
	LSTM	54.21 / 68.00	68.02 / 85.65
MC–Santander DSO	Naive-week	19.21 / 24.07	–
	DSO market	14.75 / 18.20	–
	Ridge	11.88 / 14.95	11.25 / 14.05
	LGBM	13.57 / 17.33	2.57 / 6.31
	LSTM	20.44 / 25.59	20.40 / 25.86
SIN	Naive-week	243.51 / 312.36	–
	SIN operator	259.79 / 306.48	–
	Ridge	171.61 / 220.99	141.62 / 182.38
	LGBM	204.56 / 258.12	40.12 / 96.99
	LSTM	302.80 / 373.44	357.30 / 449.74

TABLE III: Test MAPE (%) under the leakage-safe protocol (Safe) vs. full leak (Leaky-F) across datasets. Δ denotes Safe minus Leaky-F (percentage points).

Dataset	Model	Safe	Leaky-F	Δ (pp)
MC–Antioquia DSO	Naive-week	4.45	–	–
	DSO market	2.99	–	–
	Ridge	2.13	25.57	−23.44
	LGBM	2.91	12.86	−9.95
	LSTM	5.10	33.99	−28.89
MC–Santander DSO	Naive-week	5.76	–	–
	DSO market	3.85	–	–
	Ridge	3.64	19.90	−16.26
	LGBM	4.16	14.85	−10.68
	LSTM	6.34	24.32	−17.98

SIN	Naive-week	3.13	–	–
	SIN operator	2.71	–	–
	Ridge	1.93	11.17	−9.24
	LGBM	2.30	7.14	−4.83
	LSTM	3.37	13.04	−9.67

TABLE IV: Test MAE/RMSE (MW, across the 24-hour horizon) under the leakage-safe protocol (Safe) vs. full leak (Leaky-F) across datasets. Each cell reports MAE / RMSE.

Dataset	Model	Safe (MAE / RMSE)	Leaky-F (MAE / RMSE)
MC–Antioquia DSO	Naive-week	40.27 / 52.58	–
	DSO market	34.18 / 44.97	–
	Ridge	22.36 / 30.78	270.77 / 289.92
	LGBM	30.74 / 39.41	136.12 / 158.30
	LSTM	54.21 / 68.00	355.38 / 363.92
MC–Santander DSO	Naive-week	19.21 / 24.07	–
	DSO market	14.75 / 18.20	–
	Ridge	11.88 / 14.95	64.94 / 71.25
	LGBM	13.57 / 17.33	48.48 / 56.00
	LSTM	20.44 / 25.59	78.05 / 81.47
SIN	Naive-week	243.51 / 312.36	–
	SIN operator	259.79 / 306.48	–
	Ridge	171.61 / 220.99	993.09 / 1114.25
	LGBM	204.56 / 258.12	632.81 / 767.13
	LSTM	302.80 / 373.44	1153.85 / 1272.40

Future work

– Extend the evaluation beyond the fixed day-ahead setting by varying the operational gap g, horizon H, and issuance schedule.

– Report operationally aligned metrics beyond MAPE, including per-hour and per-day MAPE and exceedance rates relative to regulation-defined error limits (hourly and daily).

REFERENCES

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," International Journal of Forecasting, vol. 32, no. 3, pp. 914–938, 2016, 10.1016/j.ijforecast.2015.11.011

[2] S. Fan and R. J. Hyndman, "Short-term load forecasting based on a semi-parametric additive model," IEEE Transactions on Power Systems, vol. 27, no. 1, pp. 134–141, 2012, 10.1109/TPWRS.2011.2162082

[3] S. Ben Taieb and R. J. Hyndman, "A gradient boosting approach to the kaggle load forecasting competition," International Journal of Forecasting, vol. 30, no. 2, pp. 382–394, 2014, 10.1016/j.ijforecast.2013.07.005

[4] Comisión de Regulación de Energía y Gas (CREG), "Resolución CREG 025 de 1995 [CREG resolution 025 of 1995]," 1995, [Online]. Available: https://gestornormativo.creg.gov.co/gestor/entorno/docs/resolucion_creg_0025_1995.htm

[5] Comisión de Regulación de Energía y Gas (CREG), "Resolución CREG 100 de 2019 [CREG resolution 100 of 2019]," 2019, [Online]. Available: https://gestornormativo.creg.gov.co/gestor/entorno/docs/resolucion_creg_0100_2019.htm

[6] Unidad de Planeación Minero Energética (UPME), "Generation Projects Registry – July 2024," Jul. 2024, [Online]. Available: https://www1.upme.gov.co/siel/Inscripcion_proyectos_generacion/Registro_julio_2024.pdf

[7] XM Compañía de Expertos en Mercados S.A. E.S.P., "En 2024 se declararon en operación 67 proyectos de generación y 32 de transmisión [In 2024, 67 generation projects and 32 transmission projects entered operation]," 2025, [Online]. Available: https://www.valoraanalitik.com/cuantos-proyectos-de-energia-en-colombia-entraron-en-2024/

[8] L. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," International Journal of Forecasting, vol. 16, no. 4, pp. 437–450, 2000, 10.1016/S0169-2070(00)00065-0

[9] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," Computational Statistics & Data Analysis, vol. 120, pp. 70–83, 2018, 10.1016/j.csda.2017.11.003

[10] M. López de Prado, Advances in Financial Machine Learning. Hoboken, NJ: Wiley, 2018.

[11] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on European data," IEEE Transactions on Power Systems, vol. 22, no. 4, pp. 2213–2219, 2007, 10.1109/TPWRS.2007.907583

[12] Q. Zhao, X. Liu, and J. Fang, "Extreme gradient boosting model for day-ahead STLF in national level power system: Estonia case study," Energies, vol. 16, no. 24, p. 7962, 2023, 10.3390/en16247962

[13] P. Netzell, H. Kazmi, and K. Kyprianidis, "Deriving input variables through applied machine learning for short-term electric load forecasting in Eskilstuna, Sweden," Energies, vol. 17, no. 10, p. 2246, 2024, 10.3390/en17102246

[14] S. Kaufman, S. Rosset, and C. Perlich, "Leakage in data mining: Formulation, detection, and avoidance," ACM Transactions on Knowledge Discovery from Data, vol. 6, no. 4, pp. 1–21, 2012, 10.1145/2382577.2382579

[15] P. Subotić, U. Bojanić, and M. Stojić, "Statically detecting data leakages in data science code," in Proceedings of the 11th ACM SIGPLAN International Workshop on the State Of the Art in Program Analysis (SOAP '22). Association for Computing Machinery, 2022.

[16] S. Albelali and M. Ahmed, "Hidden leaks in time series forecasting: How data leakage affects LSTM evaluation across configurations and validation strategies," arXiv:2512.06932, 2025, arXiv:2512.06932

[17] T. Talagala, "tsdataleaks: An R package for detecting data leaks in forecasting competitions," 2024, https://cran.r-project.org/package=tsdataleaks

[18] W. Feng, R. Tao, J. Cartlidge, and J. Zheng, "Vmdnet: Time series forecasting with leakage-free samplewise variational mode decomposition and multibranch decoding," arXiv:2509.15394, 2025, arXiv:2509.15394

[19] M. Meyer, S. Kaltenpoth, K. Zalipski, H. Albers, and O. Müller, "Ts-arena: A pre-registered live forecasting platform," 2025, arXiv:2512.20761