Data Cleaning — 7-Step Pipeline

1,496,208 raw rows $\times$ 25 cols ▪ `src/data_cleaner.py` ▪ Sequential stages with `CleaningReport` tracking

Step	Function	Action	Detail	Rows After
1	`drop_nan_simscode`	Remove null join keys	Drops rows where `simsCode` is NaN; converts float → int → str	1,478,640
2	`exclude_utilities`	Drop OIL28SEC	100% zero readings — no signal; constant `EXCLUDED_UTILITIES`	1,469,856
3	`exclude_unmatched_buildings`	Drop codes 8, 43, 93	No matching metadata in SIMS; constant `EXCLUDED_SIMSCODES`	1,465,464
4	`apply_hard_caps`	Sensor-fault outlier removal	Per-utility caps: ELEC/HEAT/COOL 10k; GAS 50k; STEAM 1M	1,460,283
5	`impute_short_gaps`	Fill gaps ≤ 8 intervals	Groups by `(meterId, simsCode, utility)`; ffill then bfill. *112 cells filled*	1,460,283
6	`drop_dead_meters`	Remove 100% NaN meters	47 meters where every `readingValue` is NaN dropped entirely	1,391,475
7	`drop_sparse_meters`	Remove $>50\%$ NaN meters	Meters with NaN fraction above `SPARSE_THRESHOLD = 0.5`	**1,391,475**

Raw input columns (12):

`meterId`, `siteName`, `simsCode`, `utility`, `readingTime`, `readingValue`, `readingUnits`, `readingWindowSum`, `readingWindowMin`, `readingWindowMax`, `readingWindowMean`, `readingWindowStandardDeviation`

Cleaned output:

Same schema as raw input, with invalid rows removed, short gaps imputed, and unreliable meters excluded. Cached to `data/cleaned_{utility}.csv` for reuse.

Data Shape — Per Utility After Each Step

Row counts by utility type through the 7-step cleaning pipeline

Step	Electricity	Gas	Heat	Steam	Cooling	Total
Raw input	745,176	238,632	244,488	55,632	200,568	**1,496,208**
1. drop NaN keys	733,464	237,168	240,096	55,632	200,568	1,478,640
2. exclude utilities	733,464	237,168	240,096	55,632	200,568	1,469,856
3. exclude buildings	729,072	237,168	240,096	55,632	200,568	1,465,464
4. hard caps	727,582	237,036	239,868	54,698	199,635	1,460,283
5. impute gaps	727,582	237,036	239,868	54,698	199,635	1,460,283
6. dead meters	689,518	232,644	228,156	48,842	190,851	1,391,475
7. sparse meters	689,518	232,644	228,156	48,842	190,851	**1,391,475**
Rows removed	55,658	5,988	16,332	6,790	9,717	**104,733**
Retention	**92.5%**	**97.5%**	**93.3%**	**87.8%**	**95.2%**	**93.0%**

Biggest drop: Step 6 (dead meters)

- 47 meters removed (68,808 rows)
- Sensors completely offline during collection period

Overall: 93% data retention

- Gas best preserved (97.5%)
- Steam most affected (87.8%) — smaller sample

XGBoost Models — All 5 Utilities

ELECTRICITY ▪ GAS ▪ HEAT ▪ STEAM ▪ COOLING

INPUT FEATURES (25)

Weather (8)

- temperature_2m
- relative_humidity_2m
- dew_point_2m
- direct_radiation
- wind_speed_10m
- cloud_cover
- apparent_temperature
- precipitation

Building (3)

- grossarea
- floorsaboveground
- building_age

Temporal (4)

- hour_of_day
- minute_of_hour
- day_of_week
- is_weekend

ENGINEERED FEATURES (10)

Lag Features (4)

- energy_lag_4 — 1 h
- energy_lag_24 — 6 h
- energy_lag_96 — 24 h
- energy_lag_672 — 1 wk

Rolling Statistics (4)

- rolling_mean_96 — 24 h μ
- rolling_std_96 — 24 h σ
- rolling_mean_672 — 1 wk μ
- rolling_std_672 — 1 wk σ

Interactions (2)

- temp_x_area — $T \times A$
- humidity_x_area — $RH \times A$

TARGET: energy_per_sqft

Temporal split: Sep 2025 train / Oct 2025 test

HYPERPARAMETERS

n_estimators	1000
max_depth	7
learning_rate	0.05
subsample	0.8
colsample_bytree	0.8
min_child_weight	5
reg_alpha (L1)	0.1
reg_lambda (L2)	1.0
tree_method	hist
eval_metric	rmse
early_stopping	50 rounds
random_state	42

XGBoost Validation Metrics — All 5 Utilities

Test set: Oct 2025 (temporal hold-out) ▪ Target: energy_per_sqft

Metric	Electricity	Gas	Heat	Steam	Cooling
R^2	**0.9537**	**0.6539**	**0.9202**	**0.9646**	**0.9656**
RMSE	5.58e-5	9.49e-5	3.26e-5	2.88e-3	3.62e-4
MAE	1.32e-5	3.69e-5	1.73e-5	8.34e-4	7.14e-5
Trees Used	170	166	106	195	312
Test Samples	781,716	432,280	380,968	72,708	253,212

Notes:

- All metrics on temporal hold-out (Oct 2025); models trained on Sep 2025
- RMSE & MAE in energy_per_sqft units (vary by utility scale)
- Trees Used = actual trees after early stopping (max 1000)

Key takeaways:

- 4 of 5 utilities achieve $R^2 > 0.92$
- **Cooling** & **Steam** highest (0.97)
- **Gas** weakest fit (0.65) — noisier signal

LSTM Gas Model — Dual-Branch Architecture

GAS ▪ 1.39M params ▪ 115 buildings ▪ Test $R^2 = 0.9723$

TEMPORAL FEATURES (28)

Weather (8)

- temperature_2m
- relative_humidity_2m
- dew_point_2m
- direct_radiation
- wind_speed_10m
- cloud_cover
- apparent_temperature
- precipitation

Engineered (20)

- lag features (1h, 6h, 24h, 1wk)
- rolling mean/std (24h, 1wk)
- cross-utility interactions
- temperature × area

STATIC FEATURES (3)

• grossarea	sqft
• floorsaboveground	count
• building_age	years

TARGET: energy_per_sqft

LSTM BRANCH

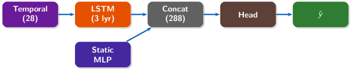
hidden_size	256
num_layers	3
dropout	0.3
bidirectional	False
max_grad_norm	1.0
seq_length	48 (12 hrs)
stride	4 (1 hr)

STATIC MLP

hidden_dims	[64]
embedding_dim	32
dropout	0.3

FUSION HEAD

head_dims	[128, 64]
activation	GELU
dropout	0.3
input_dim	288 (256+32)



TRAINING CONFIG

epochs	100 (max)
optimizer	AdamW
learning_rate	1e-3
weight_decay	1e-4
batch_size	512
scheduler	cosine
early_stop	15 epochs
normalize	z-score
wall_clock	33 min (1995s)
gpu	Tesla T4

Test $R^2 = 0.9723$

Best epoch 35 ▪ Early stop 73

Train: Sep 2025 | Test: Oct 2025

250K train ▪ 340K test samples

Gas Model Comparison — 4 Architectures

GAS utility ▪ Same data split (Sep/Oct 2025) ▪ Tesla T4 GPU

Metric	CNN	LSTM	Transformer	XGBoost
Test R^2	**0.6237**	**0.9723**	**0.9029**	**0.6539**
Training Time	8.8 min (525s)	33.3 min (1995s)	35.9 min (2153s)	1.7 min (102s)
Parameters	195K	1.39M	559K	166 trees
Model Size	2.3 MB	5.4 MB	6.5 MB	151 KB

Best accuracy: LSTM ($R^2 = 0.97$)

- Captures long-range temporal dependencies
- Dual-branch architecture leverages static features
- 3.8× better than XGBoost on gas data

Fastest training: XGBoost (1.7 min)

- 20× faster than neural approaches
- Smallest model (151 KB vs 5.4 MB)
- Lower R^2 suggests gas needs sequential modeling