Optimizing Network Resource Management via Beam-Level Traffic Forecasting with GBDTs

AI and Machine Learning in Communication Networks Workshop9th July, 2025

S **by Stephen Kolesh**

Abstract

This presentation introduces a machine learning framework for high-resolution downlink traffic forecasting in 5G networks, with a specific focus on beam-level granularity. We explore the use of Gradient Boosting Decision Trees (LightGBM and CatBoost) to model spatio-temporal traffic dynamics using telemetry features such as PRB utilization, user count, and temporal markers. A dual-pipeline strategy is proposed to handle both short-term and long-term forecasting horizons. Our approach outperforms deep learning baselines in accuracy and inference efficiency. The results demonstrate the potential of interpretable, lightweight models to support real-time traffic management and intelligent resource allocation in next-generation mobile networks.

Made with GAMMA

The Challenge of Beam-Level Traffic Forecasting

Why This Matters

Next-generation networks (5G and beyond) rely on **beamforming** to support ultra-dense user environments. Efficient **traffic forecasting at the beam level** is essential for load balancing, congestion mitigation, and QoS assurance.

The Bottle Neck

Traditional forecasting focuses on **cell or base station levels** — too coarse for modern beam-level resource scheduling.

The Opportunity

Beam-level prediction offers **granular control** but introduces:

- **High-resolution, high-dimensional data**
- **Strong temporal and spatial heterogeneity**
- **Long-term vs. short-term prediction tradeoffs**

Research Question

Can lightweight, interpretable ML models accurately predict beam-level traffic in realistic 5G scenarios — better than current baselines?

Made with GAMMA

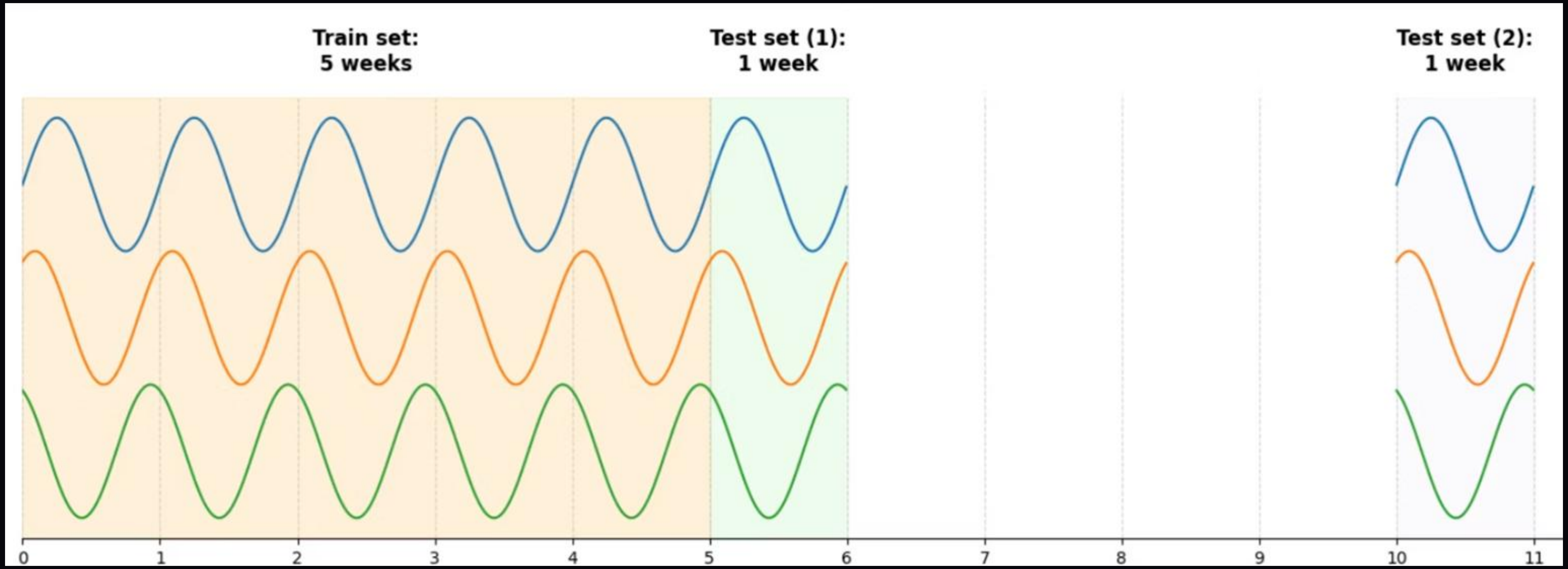
Objective

The objective of this challenge was to develop a multivariate time series forecasting model for traffic volume (DLThpVol) at the beam level.

Specifically: **Given 5 weeks of training data**, accurately forecast the behavior of the time series during:

Test set (1): Week 6 (immediate short-term prediction)

Test set (2): Week 11 (longer-horizon prediction with a 4-week observation gap)



Made with GAMMA

Given Deep Learning Baselines

Target	Hist. Avg	iTransformer	PatchTST	DLinear	Transformer
Week(6) Short Term	0.2108	0.1967	0.1973	0.2005	0.2166
Week (11) Long Term	0.2431	0.2348	0.2343	0.2352	0.2331

Made with GAMMA

Leveraging Gradient Boosting Decision Trees

GBDT Models

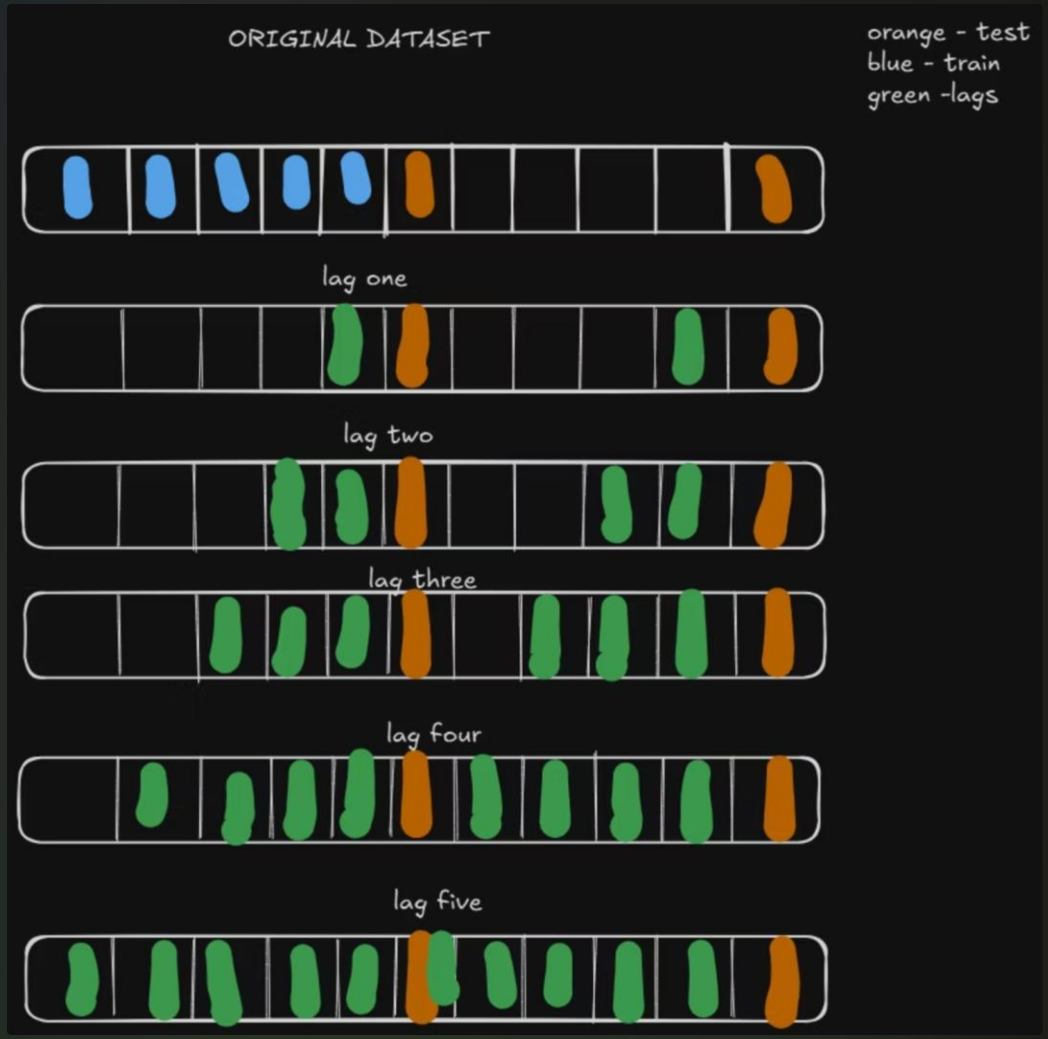
Gradient Boosting Decision Trees (GBDT) models, such as CatBoost and LightGBM, are well-suited for capturing multivariate time series dependencies.

Feature Selection

Features such as Physical Resource Block (PRB) utilization, user count, and temporal variables are used to enhance prediction accuracy.

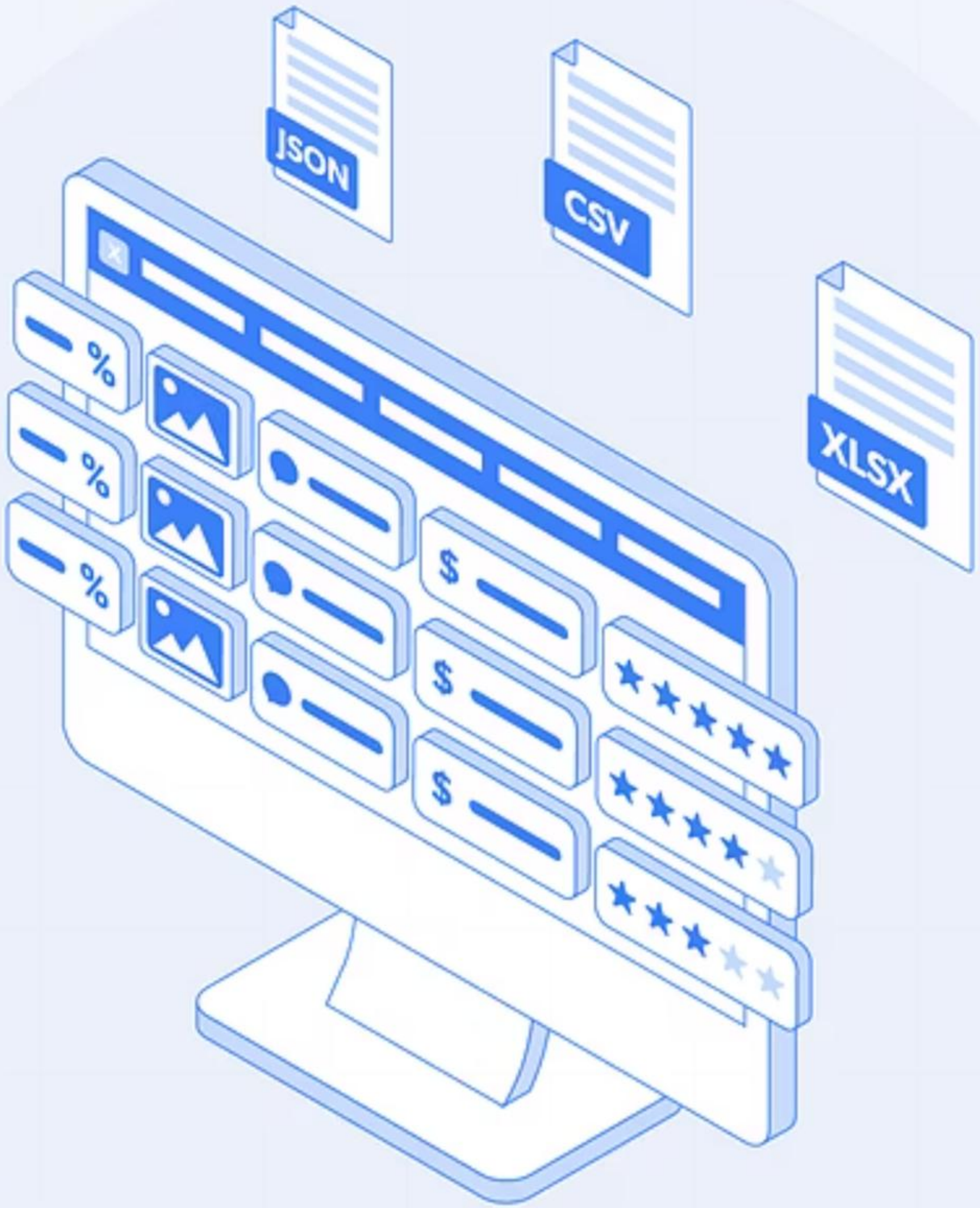
Forecasting Horizons

The model predicts traffic volumes for two horizons: one week ahead and six weeks ahead, requiring adaptability across varying timeframes.



Dataset Overview

- traffic_DLThpVol.csv: Downlink throughput volume (DLThpVol)
- traffic_DLThpTime.csv: Throughput time
- traffic_MR_number.csv: Number of users connected to each beam
- traffic_DLPRB.csv: Physical Resource Block (PRB) utilization



Feature Engineering: Enriching the Dataset

Rolling Features

> *Purpose:* Capture recent trends or volatility in the signal.
>
> For example, computing a rolling mean or standard deviation over the past 24 or 168 time steps (1 day or 1 week) helps the model understand **short-term dynamics**, which is especially important for predicting **Test Week 6**, where the model needs to rely heavily on **recent history** from Week 5.

Expanding Features

> *Purpose:* Capture long-term trends from the start of the series up to the current time.
>
> These features help the model **understand global growth or decay** in the signal. This is particularly useful for **Test Week 11**, where the model needs to **generalize across the 4-week gap** and rely on cumulative behavior rather than just recent history.

Weekly Aggregation Features

> *Purpose:* Embed weekly seasonality and past weekly behavior.
>
> By aggregating key statistics (e.g., mean, max) for each prior week, these features summarize **repeating weekly cycles** that can inform both **short-term** and **long-term** forecasts. For instance, knowing that Week 4 had a usage spike every Monday may help predict similar spikes in future weeks.

Made with GAMMA

Feature Engineering (ContD)

Shifted Features

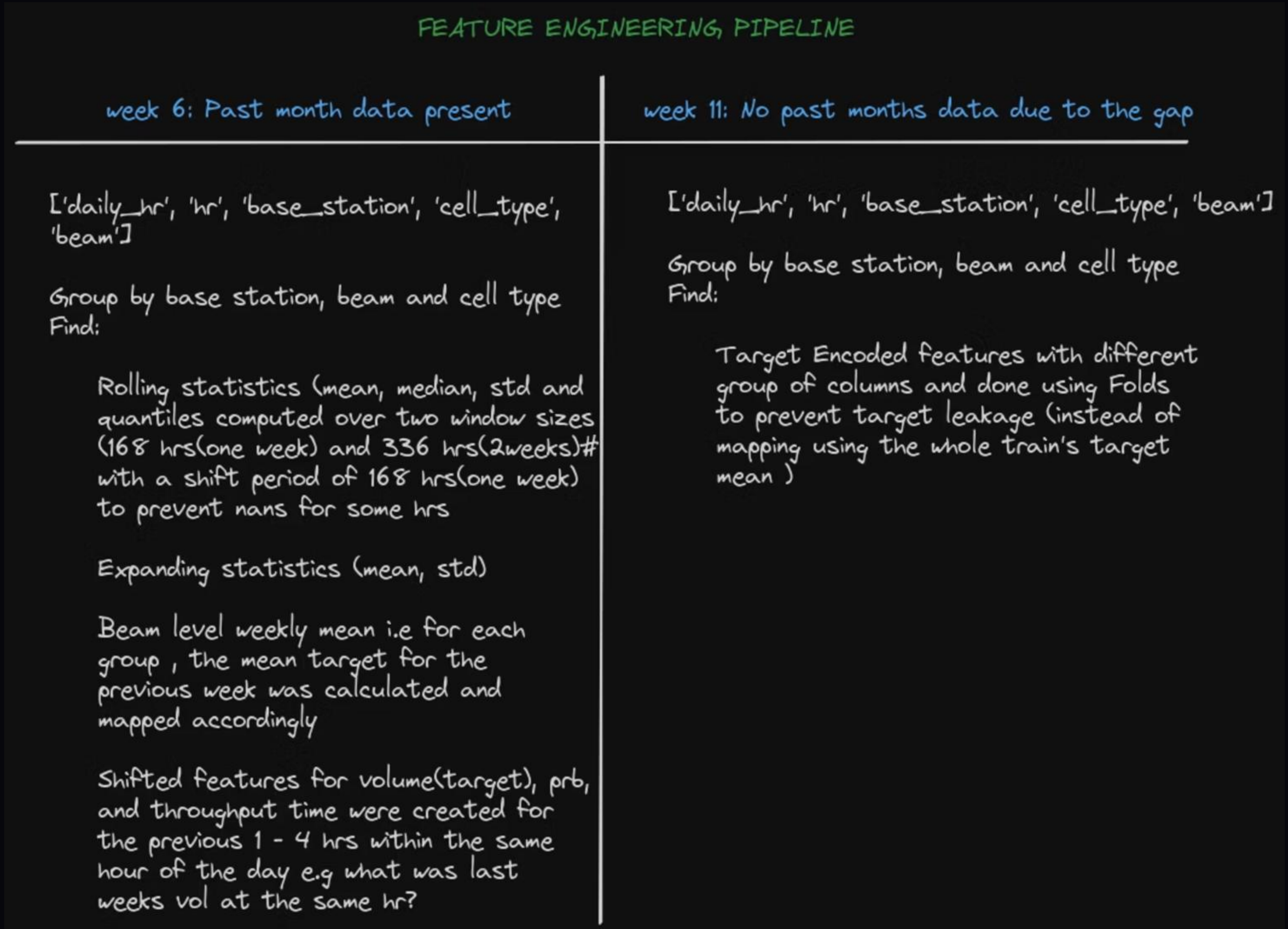
> *Purpose:* Provide explicit access to lagged values.
>
> These features let the model learn **autoregressive dependencies** — e.g., that traffic at time `t` often resembles traffic at `t-24` or `t-168`. This is essential for capturing **temporal autocorrelation**, enabling the model to learn daily/weekly **recurrence patterns** across the train and test windows.

Target Encoding

> *Purpose:* Encode categorical variables (e.g., beam ID, region, time of day) based on the target variable' s behavior.
>
> For example, encoding each beam with its average traffic volume helps the model quickly learn that **some beams or regions consistently carry more load** than others. This is especially useful for generalization — e.g., predicting **Test Week 11** for a beam that wasn't heavily active during Week 5.

Made with GAMMA

Feature Engineering Pipeline (In depth)



Made with GAMMA

Target Encoding(In depth)

Target Encoding

Group cols used:

- ['base_station']
- ['hr']
- ['daily_hr']
- ['beam']
- ['base_station', 'beam']
- ['base_station', 'hr']
- ['base_station', 'daily_hr']
- ['base_station', 'cell_type'],
- ['base_station', 'cell_type','daily_hr'],
- ['base_station', 'cell_type','beam'],
- ['base_station', 'cell_type','beam', 'daily_hr'],
- ['base_station','cell_type','beam', 'day'],
- ['base_station', 'beam','daily_hr'],
- ['beam', 'daily_hr'],
- ['beam', 'cell_type']

Target Cols Used:

- 'Target'(throughput volume)
- 'mrno' (user count)
- 'prb' (physical resource block utilization)

STatistics Calculated

Made with GAMMA

Model Training and Evaluation

Stratified K-Fold Cross-Validation

Evaluates model performance and ensures robustness by dividing the dataset into folds and training on a subset while validating on the remaining fold.

Gradient Boosting Algorithms

LightGBM and CatBoost models are used to predict traffic volume for both short-term (one week) and longer-term (six weeks) horizons.

Feature Selection Strategies

Two distinct feature selection approaches are used to optimize predictions for both time horizons, considering the challenges posed by immediate versus gap predictions.

Made with GAMMA

Pipeline : Dual Over Single Pipeline

It involves building **two separate feature generation and modeling pipelines**, one for each test window:

Short-Term Forecast Pipeline:Trained on Weeks 1–5, validated on **Week 6** (immediate next week).Focuses on **short-range dependencies** and **recency-based signals**.

Long-Term Forecast Pipeline:Trained on Weeks 1–5, validated on **Week 11**, which comes **after a 4-week gap**.Focuses on **long-range generalization**, robust to the **lack of recent data**.

Made with GAMMA

Why the Dual Pipeline is Necessary

	Week 6 Pipeline	Week 11 Pipeline
Lagged Features	Crucial (uses t-1, t-24, t-168, etc.)	Not useful (no data from Weeks 6–10)
Rolling Stats	Informative (captures recent trends)	Not available (gap in data)
Weekly Aggregates	Still valid (captures week-on-week patterns)	Still valid
Expanding Features	Useful (accumulates up to Week 5)	Very useful (generalized long-term view)
Target Encoding	Helps encode recent behavior of entities (e.g., beam load)	Crucial for generalization — captures average tendencies over history
Model Objective	Predict immediate dynamics	Predict smoothed or seasonal patterns

Made with GAMMA

Training And Evaluation Pipeline(In depth)

Training and Evaluation

Pipeline

- I trained individual models for each approach with appropriate feature selection
  - Approach One (week 11) : Used the features excluding temporal features like lags, expanding means etc
  - Approach Two (week 6) : Used all the features including the temporal features

Each individual model was used to perform inference on the week mentioned

Models

- Catboost:
- Lightgbm

Hyper Parameter Tuning

- Optuna

Training Results

- LightGbm
  - Approach one(week 11) CV : 0.19712383813106563
  - Approach two(week 6) CV: 0.1912539406395293
- Catboost
  - Approach one(week 11) CV : 0.1972058677455319
  - Approach two(week 6) CV: 0.19177680810324244

Inference Results

- LightGbm:
  - Public Leaderboard: 0.192539542
  - Private Leaderboard: 0.226230537
- Catboost:
  - Public Leaderboard: 0.191880897
  - Private Leaderboard: 0.226283792
- Ensemble:
  - Public Leaderboard: 0.191948815
  - Private Leaderboard: 0.2261804

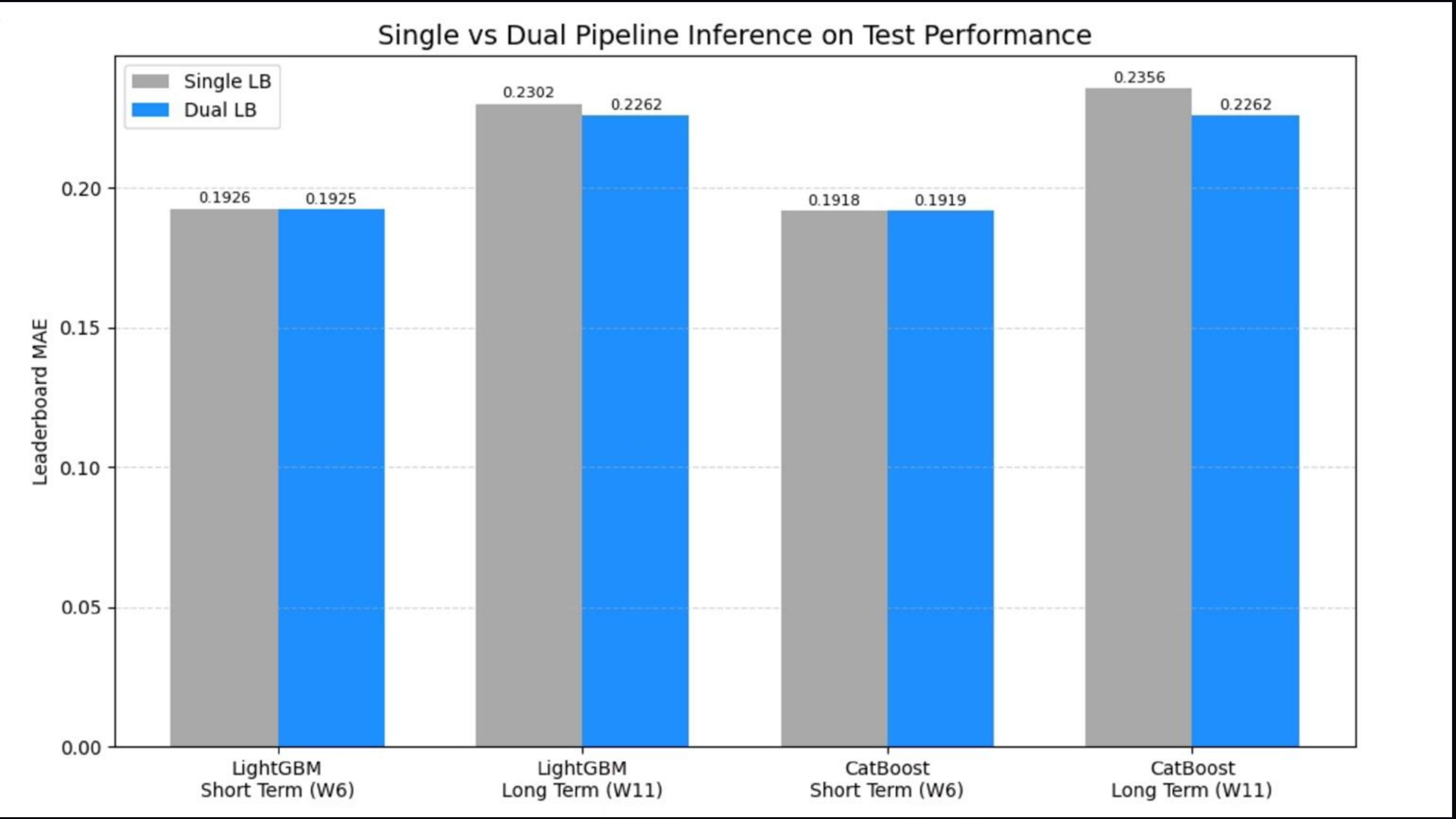
Made with GAMMA

Results

Model	Scenario	Feature Set	CV MAE	Leaderboard MAE	Δ (LB - CV)
LightGBM	Short Term(W6)	Dynamic	0.1913	0.1925	+ 0.0012
Catboost	Short Term(W6)	Dynamic	0.1918	0.1919	+ 0001
Ensemble	Short Term(W6)	Dynamic	_	0.1919	_
LightGBM	Long Term(W11)	Stable	0.1971	0.2262	+ 0.0291
CatBoost	Long Term(W11)	Stable	0.1972	0.2262	+ 0.0290
Ensemble	Long Term(W11)	Stable	_	0.2261	_

Made with GAMMA

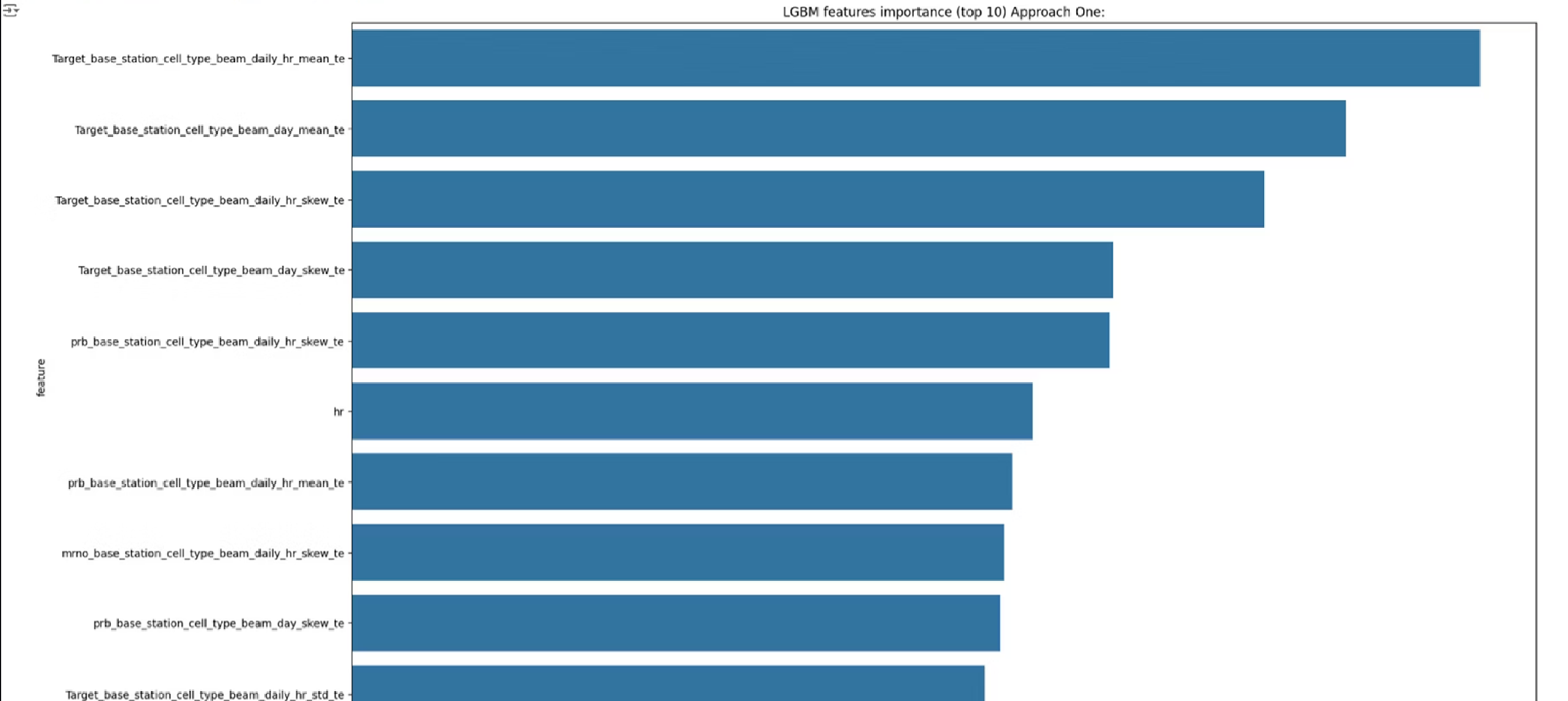
Comparison between dual and single pipeline



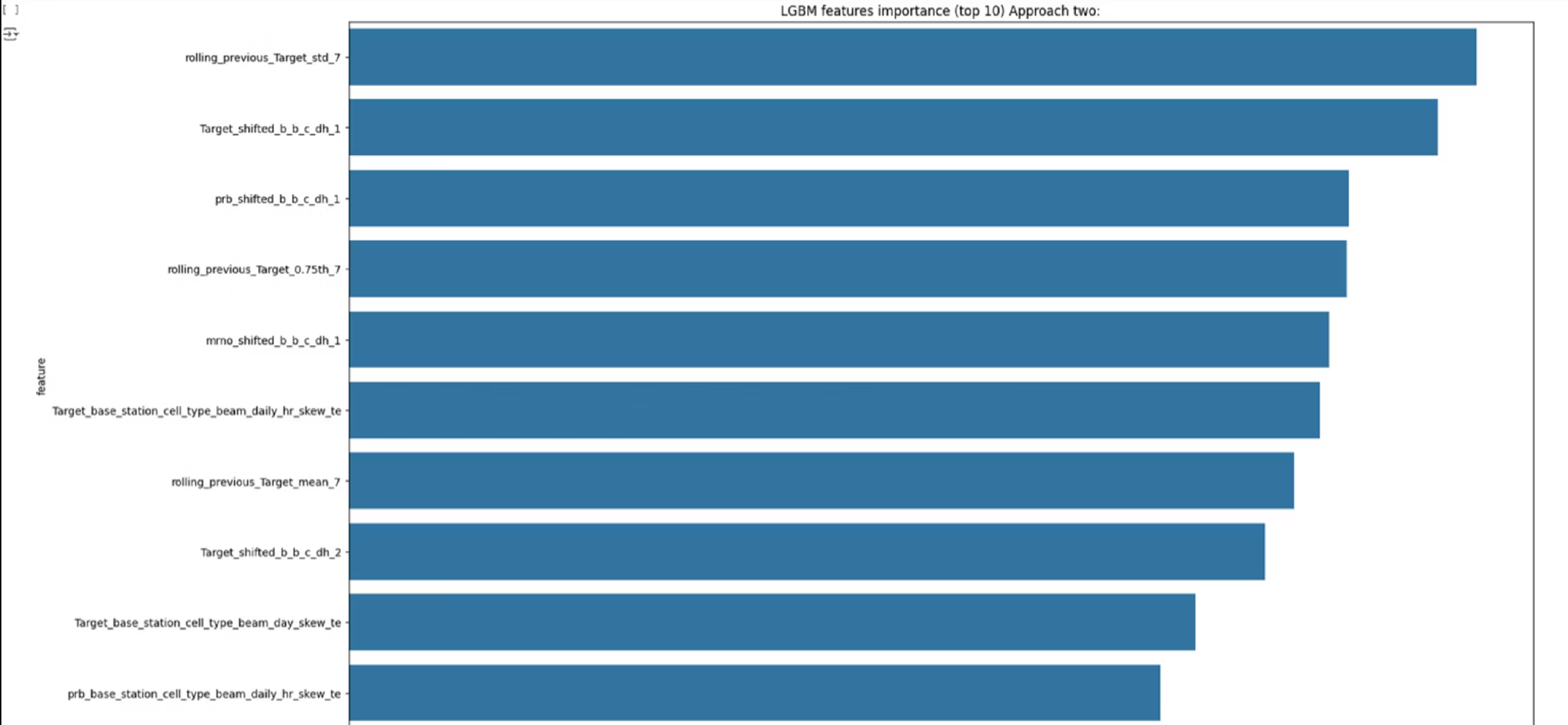
Made with GAMMA

Feature Importance (Approach One : week 11 - Long term)

Text(0.5, 1.0, 'LGBM features importance (top 10) Approach One:')



Feature Importance (Approach two : week 6 - Short Term)



Percentage improvement over benchmark models (MAE)

Benchmark Models	LightGBM W6	LightGBM W11	Catboost W6	Catboost W11	Ensemble W6	Ensemble W11
Hist. Avg	8.68%	6.95%	8.97%	6.95%	8.97%	6.99%
iTransformer	2.14%	3.66%	2.44%	3.66%	2.44%	3.71%
PatchTST	2.43%	3.46%	2.74%	3.46%	2.74%	3.50%
Dlinear	3.99%	3.83%	4.29%	3.83%	4.29%	3.87%
Transformer	11.13%	2.96%	11.40%	2.96%	11.40%	3.00%

Made with GAMMA

END

Made with GAMMA